

École Normale Supérieure de Cachan – Antenne de Bretagne
Institut de Formation Supérieure en Informatique et Communication

Pour l'obtention du :
MASTER DE RECHERCHE EN INFORMATIQUE
SPÉCIALITÉ INFORMATIQUE ET TÉLÉCOMMUNICATION

**Prise en compte de la proximité sémantique dans la
restructuration de réseau logique pair à pair**

Soutenu par :
Yann Busnel

Sous la direction d'Anne-Marie Kermarrec – Projet PARIS
Au sein de l'Institut de Recherche en Informatique et Systèmes Aléatoires



Centre National de la Recherche Scientifique
(UMR 6074) Université de Rennes 1 – Insa de Rennes

Institut National de Recherche en Informatique
et en Automatique – unité de recherche de Rennes

Prise en compte de la proximité sémantique dans la restructuration de réseau logique pair à pair

Yann Busnel

Résumé

De nombreuses approches permettant de construire un réseau reposant sur le paradigme pair-à-pair ont récemment émergé, formant des systèmes distribués de grande taille et améliorant la performance de ces réseaux à grande échelle. Une grande partie de ces réseaux logiques sont optimisés en fonction de la localité géographique des nœuds dans le réseau physique.

La prise en compte d'aspects sémantiques dans les systèmes pair-à-pair est actuellement un axe de recherche très actif. Cependant, la localisation de celle-ci au sein d'un système pair-à-pair se situe la plupart du temps dans une couche supérieure au réseau logique recouvrant le réseau physique. L'objectif de ce stage est d'optimiser la construction de ce réseau logique pair-à-pair en y intégrant directement les centres d'intérêts des utilisateurs. Pour cela, nous devons exploiter les liens existants relatifs à la sémantique d'une application lors de la construction ou la réorganisation du réseau recouvrant.

Dans ce rapport, nous introduisons les différents mécanismes de construction existants ainsi que nos travaux de recherche sur une mesure adéquate de proximité sémantique permettant de capturer plusieurs facteurs. Nous présentons également les résultats de simulations obtenus par l'exploitation de celle-ci dans un système de partage de fichiers.

Remerciements

Je tiens tout d'abord à remercier vivement les différents membres composant ce jury. Un remerciement plus particulier à Luc Bougé et Claude Jard pour la confiance qu'ils m'ont apportée depuis plusieurs années.

Je remercie chaleureusement Anne-Marie Kermarrec pour son encadrement efficace, sa confiance ainsi que sa joie de vivre. La poursuite potentielle de mon cursus en thèse sous sa direction est un honneur et un avantage que je ne saurai lui communiquer.

Je remercie également vivement Thierry Priol pour m'avoir accepté au sein de son équipe afin de valider mon stage de Master, ainsi que tout le personnel rattaché au Projet PARIS pour leur convivialité et leur gentillesse. Je tiens à remercier plus particulièrement Étienne Rivière pour sa collaboration à mes travaux, ses précieux conseils et sa jovialité.

Je tiens à remercier Fabrice Le Fessant pour sa disponibilité concernant les traces utilisées.

Enfin, je tiens à remercier affectueusement tous mes proches, et plus particulièrement Flavie Balluais, pour leur patience, leurs nombreuses relectures et leur soutien.

Mots-clés

Réseaux pair-à-pair, réseaux recouvrant, partage de fichiers, réseaux structurés et non-structurés, localité d'intérêt, mesure de distance, proximité et profil sémantique, optimisation thématique, réseaux dynamiques.



Table des matières

Résumé / Remerciements	1
Introduction	7
1 Étude bibliographique	9
1.1 Réseaux pair-à-pair	9
1.1.1 L'approche small-world	9
1.1.2 Réseaux structurés	10
1.1.3 Réseaux non-structurés	12
1.2 Localité géographique	13
1.3 Sémantique dans les réseaux pair-à-pair	15
1.3.1 Détection implicite de la sémantique	15
1.3.2 Détection explicite de la sémantique	17
1.4 Récapitulatif	18
2 Contribution	21
2.1 Motivation	21
2.1.1 Protocole épidémique	21
2.1.2 Mesure de proximité sémantique	23
2.2 Prise en compte des biais	26
2.2.1 Générosité des nœuds	26
2.2.2 Popularité des fichiers	28
3 Expérimentation	31
3.1 Traces disponibles	31
3.2 Développement d'un simulateur	32
3.3 Évaluation	34
3.3.1 Taux de succès	34
3.3.2 Prise en compte de la générosité	35
3.3.3 Prise en compte de la popularité	37
3.3.4 Évaluation de $\xi_A(B)$	39
Conclusion et perspectives	41
A Glossaire	43
Bibliographie	45

Table des figures

1	Exemple de réseau pair-à-pair	7
1.1	Principe du <i>small-world</i>	10
1.2	Table de routage dans Pastry	11
1.3	Routage dans Pastry	12
1.4	Espace de nommage dans CAN	12
1.5	Exemple d'exécution du protocole T-Man	14
1.6	Optimisation géographique dans Pastry	15
1.7	Réseau recouvrant sémantique	17
1.8	Semantic Small-World	18
2.1	Protocole de réorganisation épidémique	22
2.2	Intersection de caches	24
2.3	Générosité des nœuds	24
2.4	Popularité des fichiers	25
2.5	Relation d'ordre sémantique orientée générosité	27
2.6	Poids de λ en fonction de la valeur de γ	29
3.1	Structure de donnée type de la trace eDonkey	32
3.2	Diagramme de classe du simulateur	33
3.3	Taux de succès	34
3.4	Répartition de la charge des nœuds	36
3.5	Distribution cumulative (CDF)	36
3.6	Influence de α et β sur la CDF	37
3.7	Taux de succès en fonction de la popularité des fichiers	38
3.8	Taux de succès pour des fichiers rares	39

Introduction

Les systèmes pair-à-pair (*Peer-to-peer* ou P2P) constituent une plate-forme récente pour exécuter des applications réparties dans des environnements à large échelle. Contrairement aux approches *client-serveur*, les nœuds sont connectés entre eux directement au dessus du réseau physique (cf. figure 1). Ces systèmes permettent de partager les ressources et les services entre les différents nœuds du système qui peuvent se comporter à la fois comme client et serveur. En restant autonome, chaque ordinateur prend part au réseau global, indépendamment de son type de connectivité.

Les systèmes pair-à-pair se sont popularisés ces dernières années avec les systèmes de partage de fichiers sur Internet. Par exemple, les systèmes populaires comme Gnutella [3] et KaZaA [4] supportent des millions d'utilisateurs partageant des pétaoctets de données sur Internet. Ces systèmes sont dits non-structurés car les applications n'imposent pas de structure au réseau sous-jacent. Ils sont simples, offrent des fonctions limitées (recherche par mot-clé), et mettent en œuvre des techniques simples de recherche reposant principalement sur l'inondation du réseau. Cependant, ceci conduit à des problèmes de performance. Il existe également des systèmes P2P, dit hiérarchiques, assurant les même services mais dépendant beaucoup plus de la structure du réseau sous-jacent tels que eMule [2], eDonkey [1] ou Napster [5]. Nous nommerons ces réseaux logiques recouvrant par la suite (*overlay network*).

Cependant, les applications sur les systèmes P2P ne se limitent pas au partage de fichiers. Plusieurs domaines de recherche sont en cours d'exploration tels que la gestion de données ou le monitoring par exemple. De nombreuses fonctionnalités doivent être étudiées dans ce cadre telles que l'organisation des nœuds, la sécurité, le système de cache, la cohérence de données modifiables, la tolérance aux fautes, les algorithmes de routage et de recherche. La plupart de ces applications optimisent le réseau pair-à-pair recouvrant de manière à ce que chaque nœud choisisse ses voisins en fonction de leur proximité géographique ou de leur temps de latence par

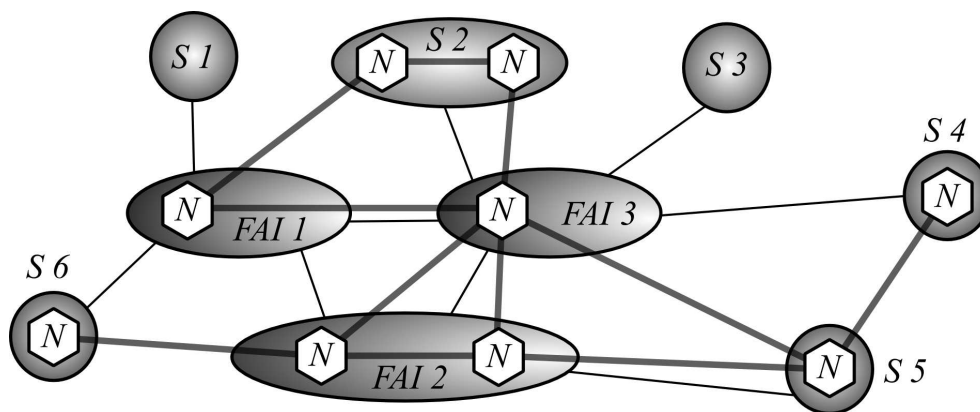


FIG. 1 – EXEMPLE DE RÉSEAU PAIR-À-PAIR AU DESSUS D'UN RÉSEAU PHYSIQUE

exemple.

Dans le cadre des applications de partage de ressources, derrière chaque machine se trouve un utilisateur. Les préférences des utilisateurs peuvent aussi être exploitées pour optimiser l'efficacité de l'application sur laquelle il se trouve. Dans le contexte de cette étude, nous étudions comment prendre en compte ces aspects que nous qualifierons de sémantiques. En effet, dans ce stage, nous cherchons à intégrer une proximité sémantique dans les systèmes pair-à-pair, avec une approche générique permettant une adaptation pour une utilisation dans différentes applications. Nous considérons dans cette approche à la fois l'optimisation des algorithmes de recherche et de l'équilibrage de charge tout en assurant la connectivité du réseau, ainsi que les algorithmes de routage et de recherche. Nous ne traitons pas des aspects de sécurité et de tolérance aux défaillances.

De nombreuses mises en œuvre existent d'ores et déjà. Cependant, pour la plupart, un choix de conception est fait du point de vue de leur localisation au sein du système. Se situant dans une couche supérieure au réseau recouvrant, elles n'interviennent pas, ou très peu, dans la construction de celui-ci. Nous proposons une alternative : remplacer les liens existants par des liens sémantiques. L'objectif de ce stage est d'intégrer les aspects de proximité d'intérêt directement dans la construction tout en maintenant la connectivité.

Afin de mieux cerner le cadre de notre étude, nous présentons dans le chapitre 1 un état de l'art étendu du domaine de notre étude. Nous développons ensuite dans le chapitre 2 les solutions choisies afin d'atteindre notre objectif et les motivations de ces choix. Dans ce même chapitre, nous introduisons la mesure de proximité sémantique développée durant ce stage. Nous présentons les résultats des expérimentations dans le chapitre 3, avant de conclure dans le dernier chapitre sur les perspectives et les problèmes restants ouverts.

Chapitre 1

Étude bibliographique

Nous développons cette étude bibliographique comme suit : la partie 1.1 présente plus en détail les différents réseaux pair-à-pair existants. Ensuite, la prise en compte de la localité géographique dans ces systèmes est traitée en section 1.2. La partie suivante introduit les différentes techniques de prise en compte de l'aspect sémantique dans ces réseaux et la partie 1.4 conclue ce chapitre par une synthèse et une présentation des objectifs de ce stage.

1.1 Réseaux pair-à-pair

Les réseaux pair-à-pair (*P2P*) génériques possèdent des capacités d'auto-organisation et de disponibilité des données. La symétrie entre les nœuds, l'équilibrage de charge et la connaissance uniquement locale du système fournissent aux réseaux P2P une capacité inhérente de passage à l'échelle.

Une analyse de ces réseaux à grande échelle sur la toile de l'Internet (*World Wide Web* ou *WWW*) est proposée dans [7]. Elle présente notamment un comparatif de divers systèmes concernant la tolérance aux fautes et l'auto-organisation. Dans cet article, parmi les systèmes introduits, on découvre le modèle théorique, dit de « petit monde » (*small-world*). Ce modèle, présenté par la suite (section 1.1.1), se révèle un modèle élégant et idéal, mais, très difficile à mettre en œuvre. A l'opposé, nous connaissons le modèle pratique dit de réseau pair-à-pair hiérarchique. Issu de l'idée de réseau à grande échelle basée sur des super-pairs contrôlant un groupe restreint de nœuds, ce modèle est effectivement plus facile à mettre en œuvre mais limite le passage à l'échelle, dû en particulier à la découverte de ressources semi-centralisée. Nous ne considérerons pas ces systèmes (eDonkey [1], eMule [2], Napster [5], etc.). Cependant, beaucoup de mécanismes étudiés par la suite s'appliquent également à cette classe intermédiaire.

Les réseaux pair-à-pair se répartissent au sein de deux grandes familles : les réseaux structurés et non structurés, présentés respectivement dans les section 1.1.2 et 1.1.3. La plupart des systèmes P2P entrent dans l'une de ces deux grandes classes, même si certains se situent à la frontière de celles-ci en définissant des systèmes hybrides [26].

1.1.1 L'approche *small-world*

Une étude sur les relations sociales a montré l'existence du phénomène de *small-world*, illustré par l'expression connue de tous : « Ah que le monde est petit ! ». En effet, Stanley Milgram et ses collègues ont présenté dans les années 60 les résultats de l'expérience décrite ci-dessous [29].

Le but était de trouver de courtes chaînes de connaissances entre deux individus aux États-Unis qui ne se connaissaient absolument pas. Dans un exemple typique de mise en œuvre de cette expérience, une personne *source* au Nebraska devait envoyer une lettre à une personne *cible* dans le Massachusetts. La source, informée uniquement du nom, de l'adresse et du métier de

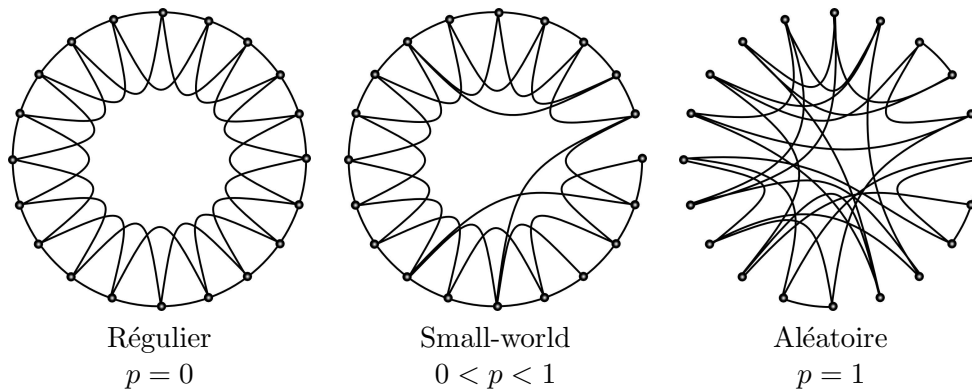


FIG. 1.1 – PROCÉDURE ALÉATOIRE DE RÉORGANISATION DE LIENS ENTRE NŒUDS POUR L'INTERPOLATION ENTRE UN ANNEAU RÉGULIER ET UN RÉSEAU ALÉATOIRE ($n = 20$ ET $k = 4$)

la cible, devait envoyer cette missive à une de ses connaissances proche (famille ou ami intime) dans l'optique d'une transmission efficace et rapide jusqu'à la cible. Quiconque recevait la lettre, obtenait les mêmes informations sur la cible, et la chaîne de transfert se poursuivait jusqu'à ce que la cible ait reçu le courrier. Après de nombreux essais, le nombre moyen d'étapes intermédiaires dans une chaîne réussie s'est avéré se situer entre cinq et six. Depuis lors, le principe des « six degrés de séparation » a connu un grand succès.

De récents travaux ont suggéré que ce phénomène serait dominant dans l'efficacité des nouveaux réseaux. En effet, un article de Watts et Strogatz dans le magazine SCIENCE [42] introduit une procédure aléatoire de réorganisation des liens entre chacun des nœuds (*Random rewiring procedure*). Le réseau de base est un anneau possédant n nœuds, chacun relié aux k nœuds les plus proches. Pour chacun des nœuds du réseau, pris dans le sens horaire de l'anneau, la procédure reconnecte ce nœud à un autre nœud choisi aléatoirement, avec une probabilité p . Ce principe est appliqué sur les n nœuds $k/2$ fois (une fois pour chacun des degrés de liaison de l'anneau). La figure 1.1 présente l'aspect du réseau après réorganisation pour trois valeurs de p différentes.

Cependant, ce modèle théorique est très difficile à mettre en œuvre de manière décentralisée en raison de sa construction et de son coût de maintenance (initialisation, dynamisme difficilement applicable dû à sa méthode de construction, etc.). De plus, malgré les excellents résultats en terme d'efficacité dans le routage des messages au sein du réseau comparativement aux structures aléatoires, ce modèle possède ses propres limites. Celles-ci sont présentées et démontrées dans [23]. Ce dernier article introduit les limitations théoriques du phénomène dit de *small-world* et donc celles du réseau proposé par [42]. En effet, Jon Kleinberg montre l'existence d'une borne inférieure et d'une borne supérieure du temps de livraison d'un message par un algorithme décentralisé, basé sur le principe des « six degrés de séparation ».

De nombreuses déclinaisons existent dans les réseaux pair-à-pair. Mais beaucoup d'autres systèmes reposant sur des modèles différents sont également apparus ces dernières années. Par la suite, nous présentons les deux grandes familles de réseaux pair-à-pair existants à ce jour ainsi que certains systèmes représentatifs de chaque famille.

1.1.2 Réseaux structurés

La première grande classe de systèmes est celle des réseaux P2P structurés. Ils sont dits structurés car au dessus du réseau physique sous-jacent, les nœuds sont reliés par un réseau recouvrant (cf. fig 1) construit sous certaines contraintes, répondant à plusieurs propriétés et connectant les pairs selon une structure particulière donnée (par exemple : un anneau, un espace

0	1	2	3	/	5	6	7	8	9	a	b	c	d	e	f
<i>x</i>	<i>x</i>	<i>x</i>	<i>x</i>	/	<i>x</i>	<i>x</i>	<i>x</i>	<i>x</i>	<i>x</i>	<i>x</i>	<i>x</i>	<i>x</i>	<i>x</i>	<i>x</i>	<i>x</i>
4	4	4	4	4	4	4	4	4	4	4	4	c	4	4	/
0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	/
<i>x</i>	<i>x</i>	<i>x</i>	<i>x</i>	<i>x</i>	<i>x</i>	<i>x</i>	<i>x</i>	<i>x</i>	<i>x</i>	<i>x</i>	<i>x</i>	<i>x</i>	<i>x</i>	<i>x</i>	/
4	4	4	/	4	4	4	4	4	4	4	4	4	4	4	4
f	f	f	/	f	f	f	f	f	f	f	f	f	f	f	f
0	1	2	/	4	5	6	7	8	9	a	b	c	d	e	f
<i>x</i>	<i>x</i>	<i>x</i>	/	<i>x</i>	<i>x</i>	<i>x</i>	<i>x</i>	<i>x</i>	<i>x</i>	<i>x</i>	<i>x</i>	<i>x</i>	<i>x</i>	<i>x</i>	<i>x</i>

FIG. 1.2 – TABLE DE ROUTAGE DANS PASTRY

cartésien, etc.).

Les réseaux P2P structurés permettent, pour la plupart, d’implémenter facilement une table de hachage distribuée décentralisée (*Distributed Hash Table* ou DHT). Nous verrons par la suite qu’ils permettent également un routage de proximité (cf. section 1.2). Nous illustrons la structure de ces réseaux dans le cadre de Pastry [34].

Pastry est un réseau recouvrant auto-organisant en anneau comme le montre la figure 1.3. Il introduit un système de routage pouvant mettre en œuvre une table de hachage totalement décentralisée, tolérante aux fautes, permettant aisément de passer à l’échelle. Ce routage assure la livraison d’un message vers n’importe quelle clé de la table. $\text{send}(n, \text{clé})$ correspond en effet à l’opération de base de ce système. Le protocole de routage est décrit ci-dessous.

Dans Pastry, chaque nœud possède un identifiant unique choisi aléatoirement dans l’espace de nommage des clés de la table de hachage, représentant sa localisation sur l’anneau. Ainsi, chaque clé est associée au nœud possédant l’identifiant le plus proche selon la relation d’ordre de l’ensemble de nommage (le plus souvent, l’ordre sur les entiers). Ce nœud est alors responsable de la valeur associée à cette clé. De plus, il maintient une table de routage ainsi qu’un ensemble de nœuds dit *voisins*. La table de routage possède typiquement $\log_{16}N$ lignes contenant 15 valeurs (si on représente les identifiants en base 16). Ces valeurs sont des identifiants de nœuds qui possèdent un préfixe de taille x avec l’identifiant du nœud considéré pour le rang x . Par exemple, le tableau de la figure 1.2 présente les trois premières lignes de la table de routage du nœud **4f35c1**.

Nous pouvons remarquer que dans chaque ligne de rang n , il n’existe pas de pointeur vers un nœud ayant un préfixe commun de taille $n + 1$. Effectivement, dans ce cas, c’est le nœud considéré qui sera son propre représentant (par exemple, pour les requêtes concernant les nœuds **4f3x**, **4f35c1** – représenté ici par / – apparaît à la troisième ligne).

Ainsi, pour chaque message, le nœud compare l’identifiant de la destination avec son propre identifiant, calcule la taille k du plus grand préfixe commun et transmet le message au nœud dont l’identifiant possède un préfixe commun de taille $k + 1$ avec celui de la destination. Dans l’exemple de la figure 1.3, **4f35c1** veut atteindre **d64fea**. Il transmet donc son message vers le nœud commençant par **dx** se trouvant dans sa table de routage, soit **d3841b**. Ce dernier fait de même mais avec une taille de préfixe supérieure ou égale à deux et transmet ainsi le message à **d61e92**. Ce processus se répète jusqu’au nœud responsable de l’identifiant correspondant à la requête (ici, **d64fe3** car plus proche de **d64fea** que **d6592b** avec la distance euclidienne).

Il peut arriver qu’une des valeurs de la table de routage soit vide, ceci étant dû au départ ou à la défaillance d’un nœud. Dans ce cas, du fait que chaque nœud connaît ses plus proches voisins, il transmet le message au nœud possédant l’identifiant le plus proche dans sa liste de voisins.

Pastry est devenu un des standards dans les systèmes pair-à-pair structurés. De nombreuses

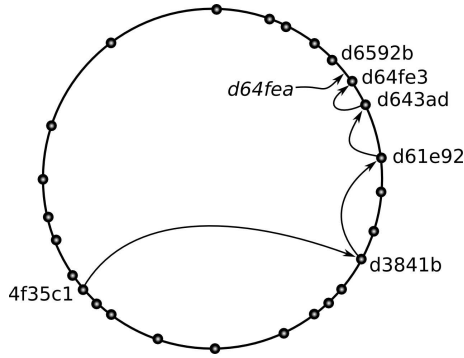


FIG. 1.3 – PASTRY : EXEMPLE DE ROUTAGE SUR L'ANNEAU UNIDIMENSIONNEL DE L'ESPACE DE NOMMAGE

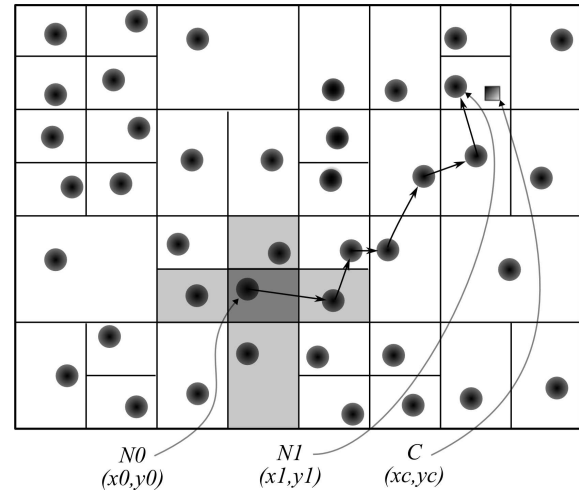


FIG. 1.4 – CAN : EXEMPLE DE ROUTAGE SUR L'ESPACE DE NOMMAGE BIDIMENSIONNEL

autres infrastructures existent telles Tapestry [43] et Chord [37] qui utilisent également une forme de routage hypercube généralisée par correspondance de préfixe. Le réseau CAN [32] est représentatif d'une autre approche utilisant un routage par progression dans un espace cartésien multidimensionnel.

Dans CAN, chaque nœud possède une localisation dans un espace cartésien multidimensionnel. Cet espace est également l'espace de nommage des clés. Pour router un message, un nœud le transmet à un de ses voisins de sorte que la distance séparant le message de sa destination soit réduite strictement. La figure 1.4 présente un exemple de routage dans CAN. Le nœud N_0 désire contacter la clé C . Pour cela, il transmet le message au nœud responsable de l'espace plus proche de sa destination. Le message transite de cette manière sur le réseau jusqu'au nœud N_1 responsable de l'espace contenant l'identifiant (x_C, y_C) de la clé C .

D'autres approches sont apparues ensuite qui prennent en compte des relations sociales afin de limiter l'action de nœuds malicieux [27, 41].

1.1.3 Réseaux non-structurés

La seconde grande famille regroupe tous les systèmes P2P non-structurés. Ceux-ci reposent sur une construction pseudo-aléatoire du graphe de connexions. Il n'existe pas d'organisation des nœuds dans un espace de nommage. Un nœud se joint au réseau par l'intermédiaire d'un autre nœud déjà connecté. Une fois inséré, le nœud sonde de manière périodique son voisinage afin de maintenir et découvrir un certain nombre de connexions. La recherche de ressources dans un tel réseau se fait selon une technique d'inondation : un nœud désirant localiser une ressource r demande à ses voisins si ils connaissent cette ressource. À leur tour, ces voisins demandent à leurs voisins si ils ont connaissance de cette ressource r et ce, jusqu'à une profondeur fixée par le système. Un nœud possédant la ressource r averti l'auteur de la requête en lui envoyant une réponse directe ou qui parcourt le chemin initial de la requête en sens inverse.

Ces réseaux permettent de développer des systèmes de partage de fichiers tels que Gnutella [3], des protocoles d'agrégation tels que des protocoles *pro-actifs* efficaces [30], ainsi que des protocoles de diffusion [15]. Certains protocoles permettent de construire ou de maintenir le réseau recouvrant tels que les protocoles SCAMP [16], LPBCAST [15], ou encore des protocoles de réorganisation des nœuds selon une heuristique particulière que nous détaillons par la suite.

A titre d'exemple, nous avons choisi une classe de protocoles permettant la construction de réseaux non-structurés. Cette classe regroupe les protocoles épidémiques (*gossip-based protocol* [15, 20, 21]) qui peuvent être décrits par un modèle généraliste [21]. Chaque nœud périodiquement met à jour sa liste de voisins en échangeant de l'information avec l'un d'entre eux choisi selon une méthode particulière. Ces protocoles épidémiques possèdent tous la même structure. Trois appels de fonctions, spécifiées dans [21] et brièvement présentés ci-dessous, permettent de les différencier.

Sélection du voisin Le choix du nœud participant à l'échange peut se faire soit aléatoirement, soit en prenant le nœud en tête de sa liste de voisins, soit en prenant le nœud en queue de celle-ci (choix effectué par la fonction *sélectionPair*).

Propagation de sa vue (ou liste de voisins) Elle peut se propager de trois façons différentes : l'envoi de sa vue au nœud sélectionné (*push*), la demande de la vue du nœud sélectionné (*pull*) ou l'échange des vues entre ces deux nœuds (*push/pull*).

Sélection de sa vue Le choix des nœuds qui composeront la liste des voisins peut se faire soit aléatoirement (sélection de c éléments de la vue), soit en prenant les c premiers nœuds apparaissant en tête de sa liste de voisins, soit en prenant les c derniers nœuds apparaissant en queue de celle-ci (choix effectué par la fonction *sélectionVoisins*).

L'étude [21] présente l'évaluation pour chaque sous-classe de la longueur moyenne d'un chemin (*average path length*), du coefficient d'agrégation (*clustering coefficient*) et du degré de connectivité moyen d'un nœud, ceci dans de nombreux cas de figure (arrivée de nœuds, stabilité du système, départ de nœuds). Nous pouvons ainsi nous rendre compte de l'efficacité de chacune des sous-classes dans telles ou telles conditions. Elle permet également de mettre en lumière leur grande tolérance aux défaillances grâce à une actualisation constante permettant de prendre en compte la dynamique des réseaux.

1.2 Localité géographique

La plupart des systèmes pair-à-pair construisent un réseau logique au dessus du réseau physique, basé sur le protocole TCP/IP, un standard dans la communication réseau. Il n'y a cependant, à priori, pas de corrélation entre les liens logiques et le réseau sous-jacent au système P2P.

Il s'avère que si le système ne prend pas en compte la topologie réseau, l'efficacité peut s'en retrouver considérablement amoindrie. En effet, cela risque d'induire un large surcoût sur les communications entre deux nœuds, une surcharge importante du réseau et une augmentation substantielle de la latence entre deux points du réseau logique. Par exemple, imaginons que A veuille communiquer avec C et que pour se faire, le réseau logique induit un passage par B . S'il n'existe pas de corrélation entre le réseau recouvrant et le réseau physique, on risque de faire transférer le message par B très éloigné de A alors que C est proche de A .

Afin d'éviter cette perte d'efficacité, il est préférable de prendre en compte la topologie réseau dans le choix des liens entre les nœuds du système. Dans les réseaux P2P non-structurés, l'absence de structure sous-jacente impose moins de contraintes concernant le choix des voisins et ainsi permet la mise en œuvre de protocoles simples, élégants et efficaces. Par exemple, il est possible de construire le réseau en prenant pour voisins, les nœuds les plus proches géographiquement, ou d'utiliser un protocole de réorganisation dans lequel la fonction *sélectionVoisins* ne conserve que les nœuds situés localement (cf. protocole de *gossip*, section 1.1.3). Nous pouvons trouver dans la communauté des articles concernant l'optimisation des réseaux recouvrants non-structurés à grand échelle comme *Localiser* [28] qui propose une solution de restructuration, d'équilibrage de charge et de tolérance aux fautes dans ce contexte.

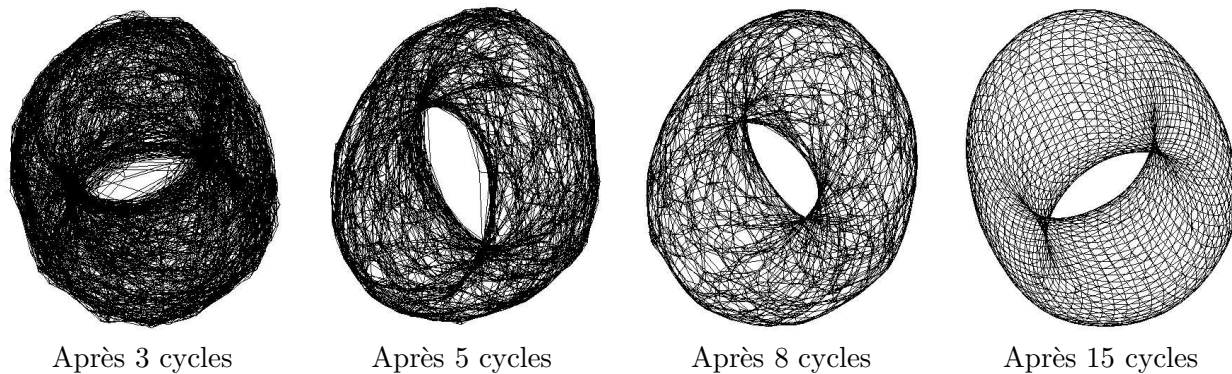


FIG. 1.5 – T-MAN : EXEMPLE ILLUSTRANT LA CONSTRUCTION D’UN TORE SUR $50 * 50 = 2500$ NŒUDS RELIÉS À 4 VOISINS

Nous allons illustrer ces mises en œuvre autour de T-Man [20], un protocole de réorganisation épidémique. Ce dernier permet de construire, à partir d’un réseau recouvrant aléatoire, une classe générale de topologie réseau. T-Man utilise deux processus, l’un passif et l’autre actif, respectant la structure d’un protocole dit de *gossip* (cf. section 1.1.3).

À chaque cycle, chaque nœud échange l’identité de ses voisins ainsi que la sienne avec le nœud le plus proche géographiquement. Il fusionne ensuite les deux listes et ne conserve que la moitié de la liste résultante en utilisant la fonction *sélectionVoisins*. Dans T-Man, *sélectionVoisins* choisit les nœuds les plus proches géographiquement. La figure 1.5 illustre l’efficacité et la rapidité de convergence de ce protocole par la construction d’un tore de 2500 nœuds. Après 5 cycles, on devine déjà la forme définitive, et après seulement 15 cycles, 99% de la topologie finale est en place.

Prendre en compte la topologie réseau dans les systèmes P2P structurés comme Pastry [34] ou CAN [32] est plus difficile, ces derniers étant contraints par la structure sous-jacente. De nombreuses études ont été faites dans ce domaine telles que [12] qui présente une sélection de voisins suivant leur proximité, et introduit deux heuristiques d’approximation pour les réseaux structurés. Nous illustrons ces études avec le cas de Pastry. Dans ce système, la localité géographique est prise en compte lors de la mise à jour de la table de routage. La probabilité de trouver un nœud proche répondant aux contraintes est plus élevée dans les lignes hautes de la table de routage, les candidats potentiels étant beaucoup plus nombreux. En représentant les nœuds sur un plan géographique et non sur l’espace de nommage comme à la figure 1.3, les routes ont la physionomie présentée dans la figure 1.6. Ainsi dans le cas de notre exemple, `d3841b` qui apparaît loin de `4f35c1` dans l’espace de nommage (cf. figure 1.3) est en réalité très proche géographiquement, et inversement pour `d64fe3` par rapport à `d643ad`.

Une évaluation [11] permet de faire le point sur toutes les études menées jusqu’à lors sur la prise en compte de la topologie dans les réseaux P2P structurés. Cet article compare les quatre grands systèmes cités précédemment [32, 34, 37, 43]. Il oriente la discussion autour des trois grandes familles de routage : par proximité (où à chaque étape de routage, le nœud le plus proche est choisi dynamiquement), par assignation des identifiants basée sur la topologie (en faisant en sorte que l’espace de nommage reflète la topologie du réseau, ce qui est difficile à assurer) et par sélection des voisins selon leur proximité (mise en œuvre en amont lors de la construction des tables de routage).

Ainsi, de nombreuses infrastructures se sont développées autour d’applications prenant en compte la localité géographique. Récemment, nous avons vu apparaître une tendance à prendre en compte un autre type de localité caractéristique aux applications utilisant de tels réseaux : la localité dite d’intérêt. La partie suivante est consacrée à ces systèmes.

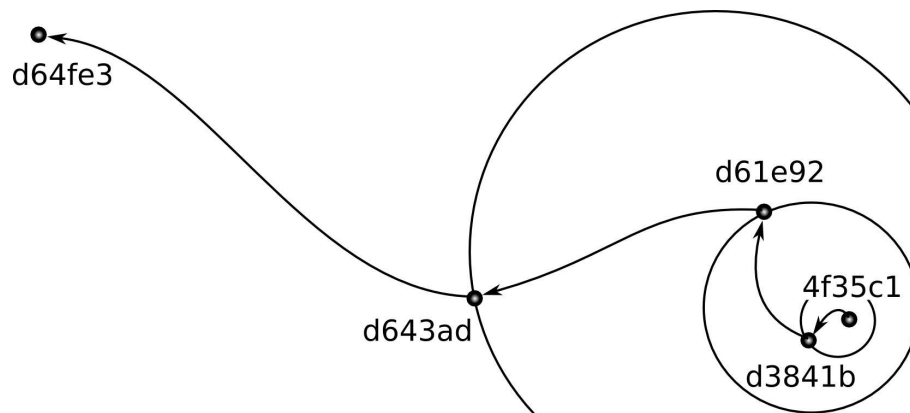


FIG. 1.6 – PASTRY : PRISE EN COMPTE DE LA PROXIMITÉ GÉOGRAPHIQUE

1.3 Sémantique dans les réseaux pair-à-pair

De nombreux travaux se penchent aujourd’hui sur des techniques similaires pour capturer non plus la localité géographique mais la localité d’intérêt ou sémantique dans les réseaux pair-à-pair. Ces systèmes exploitent des similitudes d’intérêts entre les utilisateurs d’une application donnée. De manière générale, une couche additionnelle est développée au dessus du réseau recouvrant prenant en compte ce type de localité. Des évaluations sont conduites sur ces systèmes notamment celle [17] sur différentes traces du logiciel de partage de données eDonkey2000 [1]. Les résultats de cette étude offrent une comparaison entre l’efficacité de la recherche avec une liste de voisins aléatoires et celle au sein d’un groupement de nœuds (*Clustering*) basé sur des intérêts communs.

Dans ce type de groupement thématique, l’aspect sémantique peut être exposé ou non. Les méthodes permettant de gérer la localité sémantique peuvent être classées en deux grandes familles présentées ci-dessous.

1.3.1 Détection implicite de la sémantique

La première classe regroupe les méthodes permettant de prendre en compte les aspects sémantiques des nœuds de manière implicite ou non-intrusive. Dans ce cas, les systèmes cherchent à deviner les intérêts et les préférences de chacun des utilisateurs. Le plus souvent, ces systèmes se basent sur l’utilisation d’un historique récent afin de catégoriser les nœuds du réseau P2P.

En effet, différentes techniques implicites de groupement de nœuds centrées sur la similarité des intérêts de chacun des participants ont été proposées [39]. Ainsi, une recherche orientée sémantique permet d’améliorer les chances de succès d’une requête dans les systèmes de partage de fichiers. Effectivement, les nœuds du système émettent leurs requêtes en priorité vers un ensemble de voisins privilégiés.

Cet article introduit notamment trois heuristiques de sélection de voisins décrits ci-dessous.

LRU (*Last Recently Used*) Chaque nœud possède une liste ordonnée de voisins. Pour chaque voisin répondant correctement à une requête, celui-ci est placé en tête de la liste (et le voisin en queue de liste sera éliminé si besoin est). Ainsi, par la suite, chaque requête est d’abord envoyée au voisin en tête de la liste, puis au suivant, etc.

Historique Cette méthode capitalise les nœuds sémantiquement proches sur une période plus longue. Tandis que LRU choisit les x nœuds les plus récemment utiles, la méthode historique sélectionne les x nœuds les plus utiles durant une plage de temps déterminée. Par exemple, chaque nœud depuis lequel nous avons téléchargé précédemment est associé à un

compteur. Chaque téléchargement à partir du nœud i sur le nœud j entraîne l'incrémenta-tion du compteur du nœud j . La liste des voisins correspond alors à l'ensemble des x nœuds possédant les x plus grandes valeurs de compteurs. Malheureusement, malgré l'efficacité de cette méthode dans le cas général, elle ne convient pas à un utilisateur versatile (ie. changeant de profil sémantique relativement souvent). En effet, cette méthode a un effet mémoire plus important et elle est plus longue à prendre en compte la modification du profil sémantique de l'utilisateur (le temps que les valeurs des compteurs se compensent).

Popularité Cette solution est une méthode hybride des deux précédentes. Elle permet d'obtenir des listes sémantiques contenant la plupart du temps des nœuds de même type, en conservant la simplicité de LRU. Le principe est basé sur le fait que la popularité d'un document peut influencer les résultats d'une recherche. Considérons un fichier très populaire, beaucoup de nœuds possèdent ce dernier, même s'il ne correspond pas toujours pas à leur profil sémantique. Ainsi, la popularité croissante de certains fichiers peut biaiser la proximité sémantique, en rapprochant deux nœuds n'ayant en commun que des fichiers populaires. Cette méthode permet de limiter l'effet de bord dû à la popularité.

Toutes ces méthodes permettent de capturer et d'exploiter les relations sémantiques observées dans les systèmes de partage de fichiers [1, 2, 3, 4] de manière implicite. Une évaluation des deux premières stratégies a été développée [18]. Celle-ci compare le taux de succès d'une requête pour chacune d'entre elles. En guise de point de comparaison, l'article se fonde sur une stratégie aléatoire pour le choix de voisins.

L'évaluation utilisée est la mesure de taux de succès en cas de recherche : chaque nœud de la liste est consulté en premier lieu. Puis, si le fichier n'est pas trouvé chez les voisins sémantiques, le mécanisme de base (inondation) est utilisé.

Le taux de succès de chacune des trois méthodes (aléatoire, LRU et historique) dans la recherche d'un fichier sur le réseau est très différent. En effet, pour une recherche avec une liste de voisins sémantiques de 20 nœuds, la stratégie aléatoire possède un taux de succès de 1%, alors que les méthodes LRU et historique obtiennent des taux de l'ordre de 45% (respectivement 44% et 47%).

Cette étude se penche également sur l'intérêt de la recherche à travers plusieurs niveaux de voisins. Le principe de cette méthode est simple : chaque nœud possède une liste de voisins sémantiques directs ainsi que la liste de chacun de ses voisins. Ceci permet d'illustrer le fait que « les amis de mes amis sont susceptibles d'être eux-même mes amis ! ». Cette réflexion se fonde sur la présence d'un coefficient d'agrégation non négligeable. Cela permet d'augmenter le taux de succès de 47% à 62% pour une taille de liste identique, mais il est nécessaire de pondérer ce résultat par le nombre de nœuds contactés. En effet, pour une liste contenant 5 voisins, dans la stratégie à deux niveaux, 25 nœuds sont contactés. Les résultats obtenus en contactant le même nombre de voisins sont en réalité similaires pour les deux stratégies.

D'autres études ont été développées dans le contexte de la prise en compte implicite de la localité sémantique. Par exemple, l'article [35] propose une technique de *raccourci* dans les réseaux pair-à-pair construit sur Chord [37]. Cette méthode ajoute quelques liens transversaux de la structure en anneau de Chord, permettant ainsi d'améliorer l'efficacité de la recherche dans le système. De même, d'autres cadres d'études ont été présentés. Par exemple, certains systèmes pair-à-pair utilisent les relations sociales de leurs utilisateurs afin de faire croître la confiance en un chemin et ainsi de réduire le problème de faux-routage dû à la présence de nœuds malicieux [27] ou de permettre aux utilisateurs de vérifier la bonne configuration de leurs machines par comparaison avec celles de leurs amis [41].

Cette classe de méthode est fondée sur l'idée que l'introduction implicite de la sémantique permet d'obtenir une meilleure efficacité, avec une transparence totale pour l'utilisateur. Mais ces techniques restent encore difficiles à mettre en œuvre car il n'est pas trivial de détecter cette localité sémantique de manière non intrusive.

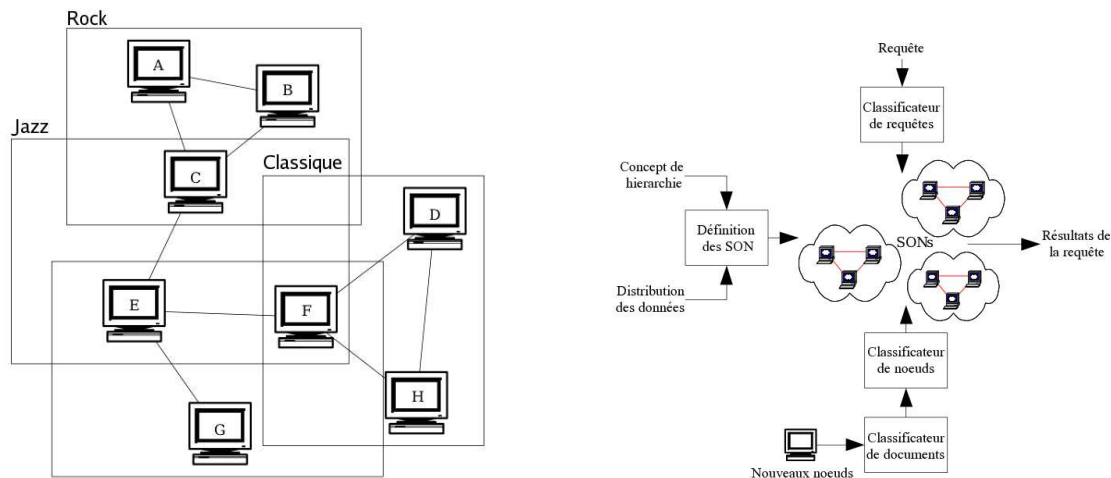


FIG. 1.7 – SEMANTIC OVERLAY NETWORK : RÉSEAU RECOUVRANT BASÉ SUR LES ASPECTS SÉMANTIQUES

1.3.2 Détection explicite de la sémantique

La seconde grande classe de méthode permettant de prendre en compte la localité d'intérêt est la classe des systèmes pair-à-pair utilisant explicitement l'aspect sémantique. Dans ces réseaux, chaque nœud doit se placer lui-même, ou donner des éléments permettant de le placer, dans une catégorie. Une telle approche, plus statique, induit une contrainte de maintenance pour l'utilisateur. Afin d'illustrer cette classe de méthode, nous allons présenter deux approches de cette catégorie.

La première d'entre elles propose un regroupement explicite de nœuds au sein du système pair-à-pair afin d'améliorer l'efficacité [13]. Celle-ci introduit le concept de réseau recouvrant sémantique (*Semantic Overlay Network* ou SON). La figure 1.7 présente le principe de ce système. Au dessus d'un réseau recouvrant existant, les nœuds sont regroupés en plusieurs SON en fonction de leurs intérêts communs. Par exemple, *A*, *B* et *C* font partie d'un même SON car chacun d'entre eux aime le *Rock*. Un nœud peut faire partie de plusieurs SON permettant ainsi à un utilisateur de ne pas se limiter à un seul thème. La partie droite de la figure 1.7 illustre le fonctionnement du réseau. Un nœud désirant rejoindre le réseau se connecte au classificateur de nœuds en fournissant son profil sémantique généré par le classificateur de documents. Les SON sont répertoriés dans le classificateur de SON qui gère la hiérarchie et la distribution des données. Une requête est fournie au classificateur de requête, avant d'être acheminée dans les différents SON correspondants. Cette méthode impose une classification statique (en ontologie par exemple) que tous sont censés connaître.

Une évolution de ce système est proposée dans [25]. En effet, tous les systèmes pair-à-pair existants traitent les aspects sémantiques dans une couche de plus haut niveau que le réseau recouvrant. Ce système propose d'intégrer la sémantique dans la construction du réseau pair-à-pair. Chaque nœud est placé selon son profil sémantique dans un modèle multidimensionnel tel que CAN [32]. Il introduit également la notion de réseau small-world sémantique (*Semantic small world* ou SSW) dans le but d'augmenter l'efficacité de la recherche basée sur l'intérêt. De plus, afin d'améliorer l'efficacité du routage dans leur système, les auteurs proposent une méthode de projection dans un domaine unidimensionnel afin de traiter le routage de la même façon que dans les systèmes de type Chord [37]. Cette projection est présentée dans la figure 1.8. Un ordre de parcours des domaines sémantiques (C_2, C_4, \dots) est créé et permet la gestion unidimensionnelle du routage. Nous pouvons observer également la présence de raccourcis (entre C_4 et C_{10} selon

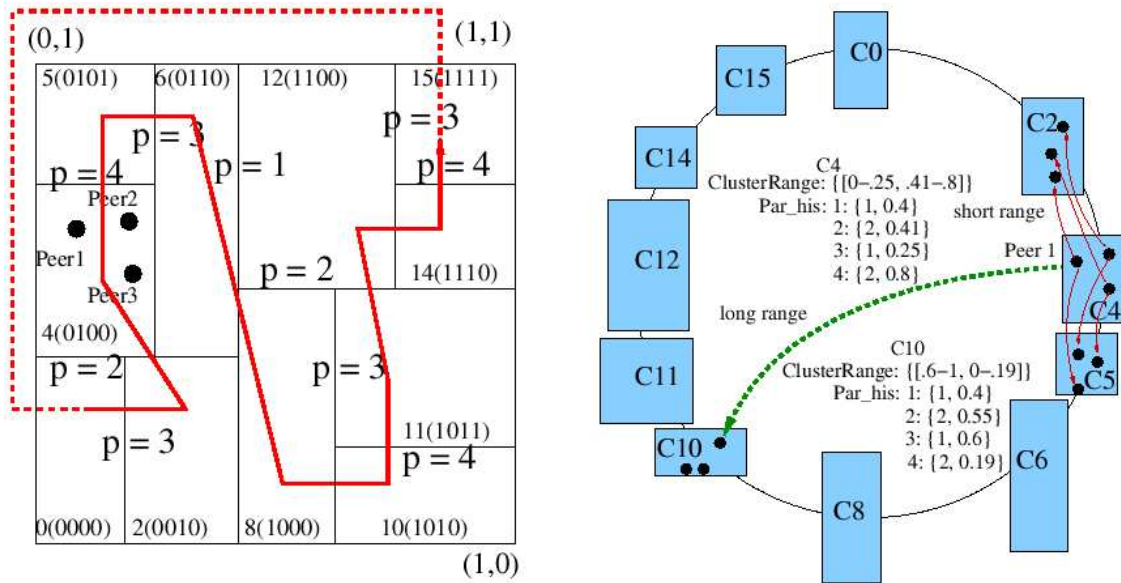


FIG. 1.8 – SEMANTIC SMALL-WORLD : UNE AMÉLIORATION DES SON PAR L'INTRODUCTION DU MODÈLE DE *small-world*

le principe introduit dans la section 1.3.1) qui permettent d'obtenir une structure de type small-world (présence de liens longue distance en plus des liens de voisinage).

D'autres solutions existent dans l'approche explicite. Notamment, une extension d'un algorithme de localisation sous-jacente permet à celui-ci de fonctionner dans un environnement décentralisé [38]. Ce système utilise un algorithme spécialisé pour retrouver l'information sur le réseau, des méthodes d'index rotatif (*rolling-index*) et d'amorçage (*bootstrapping*) pour équilibrer la charge du réseau ainsi qu'une technique d'indexation pour améliorer l'efficacité de la recherche.

Ces deux grandes classes divisent le domaine de la prise en compte des aspects sémantiques. Mais, afin de se rendre compte de la localité d'intérêt entre les nœuds, nous devons être en mesure d'évaluer les distances entre deux profils sémantiques. C'est l'un des défis de ce stage.

1.4 Récapitulatif

Cette étude bibliographique présente un aperçu du domaine dans lequel se déroule ce stage de fin d'année de Master de recherche en informatique. Nous avons introduit le concept de réseau pair-à-pair, en classifiant les différents systèmes existants en deux grandes familles : les réseaux structurés et non-structurés. Puis, nous avons présenté les différentes techniques de prise en compte des aspects sémantiques dans les réseaux pair-à-pair. Encore une fois, nous avons classifié ces méthodes en deux grandes familles : prise en compte explicite ou implicite.

Alors que la prise en compte géographique influe directement sur la structure du réseau recouvrant, les différentes approches de la prise en compte sémantique se situent pour la plupart au-dessus de celui-ci. L'un des objectifs de ce stage est l'intégration de l'aspect sémantique dans la construction et la maintenance du réseau recouvrant pair-à-pair, et non dans une couche supérieure.

Effectivement, nous nous intéressons à la problématique de recherche et de localisation de ressources et non aux aspects de transferts de données. La conception de ce système aura pour optique l'optimisation thématique du réseau ainsi que la prise en compte du dynamisme et des profils sémantiques. Pour cela, nous devons être en mesure d'évaluer la distance entre deux

profils par exemple. Pour cela, nous devons être en mesure d'évaluer la distance entre deux profils par exemple. L'objectif principal de ce stage est donc la proposition et l'étude d'une mesure de distance sémantique générique. Enfin, l'évaluation du système s'effectue dans le cadre du partage de fichiers, le projet d'accueil disposant d'une trace très importante du logiciel eDonkey2000 [1] (plusieurs gigaoctets représentant quelques semaines d'observation du réseau).

En d'autres termes, à l'image de l'aspect géographique développé actuellement, nous chercherons à appliquer l'aspect sémantique à la structure du réseau. Un des autres buts de ce stage est donc la proposition et l'étude d'une mesure de distance sémantique générique. Enfin, l'évaluation du système s'effectue dans le cadre du partage de fichiers, le projet d'accueil disposant d'une trace très importante du logiciel eDonkey2000 [1] (plusieurs gigaoctets représentant quelques semaines d'observation du réseau).

Le chapitre suivant présente nos motivations et objectifs ainsi que mes différentes contributions dans la résolution de ceux-ci.

Chapitre 2

Contribution

Ce chapitre est développé comme suit : la section 2.1 introduit nos motivations. Nous y expliquons notamment nos choix quand au modèle de protocole utilisé et aux priorités de notre étude. Cette partie introduit également les divers effets de bords et problèmes ouverts auxquels nous avons été confrontés. La présentation de nos solutions est développée dans la section 2.2.

2.1 Motivation

Comme nous l'avons introduit dans le chapitre 1, notre objectif est de prendre en compte une localité d'intérêt dans la construction du réseau recouvrant. De plus, nous avons fixé le critère de détection de la sémantique de manière non-intrusive pour les diverses raisons évoquées dans la section 1.3.1.

2.1.1 Protocole épidémique

Choix du système

Parmi les différents systèmes introduits dans le chapitre 1, il faut faire un choix permettant de réaliser notre objectif tout en conservant les caractéristiques de simplicité et d'efficacité propres aux réseaux P2P. Afin de pouvoir évaluer notre modèle sur une application existante, nous nous plaçons dans le cadre des systèmes de partage de fichiers. Nous disposons également de plusieurs traces réelles que nous présentons dans la partie 3.1.

Désireux d'obtenir un modèle simple et non contraignant, nous devons donc étudier l'intersection entre deux classes de systèmes pair-à-pair précédemment introduites : l'ensemble des réseaux pair-à-pair non-structurés dans lequel nous introduisons une détection implicite de la sémantique.

Parmi les différents systèmes de construction et de restructuration du réseau recouvrant, nous choisissons le modèle de protocole de restructuration épidémique, utilisé par exemple dans T-Man [20] et dans Vicinity/Cyclon [40]. Les raisons de ce choix sont les suivantes :

Simple Le modèle est constitué uniquement de deux processus courts et échangeant peu de données (explicités dans la partie suivante).

Efficace La figure 1.5 illustre l'efficacité de ce protocole dans la réorganisation de nœuds. L'observation de cet exemple permet de se rendre compte de la rapidité de convergence pour une heuristique donnée (dans ce cas, la localité géographique).

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

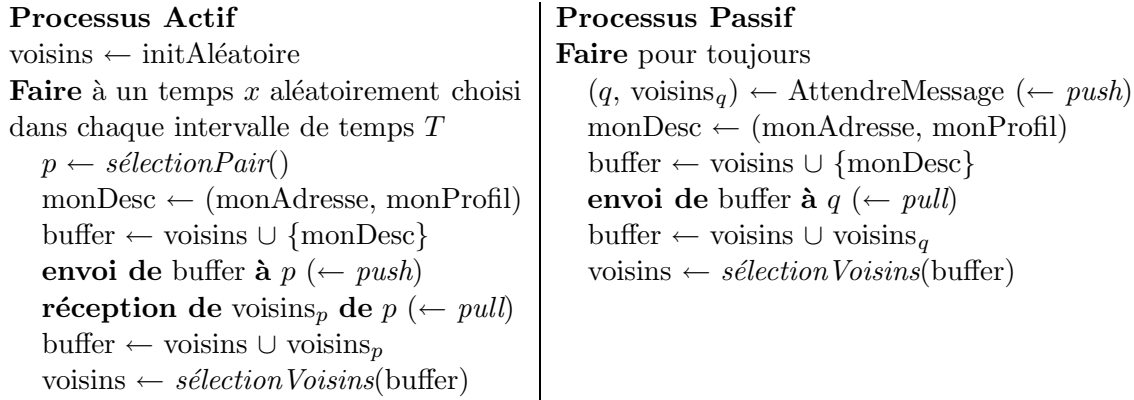


FIG. 2.1 – PROTOCOLE DE RÉORGANISATION ÉPIDÉMIQUE DES LIENS ENTRE NŒUDS

Les mots clés en italique indiquent l'appel à l'une des fonctions paramétrable des protocoles épidémiques. La présence de (\leftarrow push) ou (\leftarrow pull) sur une ligne précise la nécessité de celle-ci en fonction de la propagation choisie. Ici, nous avons représenté le schéma total dit de push/pull.

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

Localisé directement au dessus du réseau physique Ce protocole permet de réorganiser les liens entre les nœuds du réseau recouvrant, correspondant à l'un de nos objectifs.

La suite de cette partie présente de manière plus précise une modélisation des protocoles épidémiques.

Présentation du modèle

Une étude [21] citée dans la section 1.1.3, propose une modélisation des protocoles épidémiques. Celle-ci introduit une structure de base des deux processus acteurs du système. Cette structure est présentée en figure 2.1.

Le squelette du protocole peut être paramétré à l'aide des trois fonctions décrite dans cette même section 1.1.3 :

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

Dans le cas de T-Man [20] par exemple, la vue a une taille c donnée et les nœuds au sein de celle-ci sont ordonnés selon une proximité géographique décroissante. Les paramètres du protocole sont les suivants :

- Sélection du voisin : Choix d'un nœud aléatoire dans la première moitié de la vue.
- 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 une mesure de distance d choisis dans l'union des vues des deux nœuds participant à l'échange.

Cet exemple de protocole de réorganisation épidémique est à l'origine de nos motivations. Afin de pouvoir l'appliquer dans un contexte sémantique, nous devons déterminer une mesure de proximité d'intérêt entre deux nœuds du système. Celle-ci nous est indispensable compte tenu de l'ordonnement des voisins au sein de la vue et le choix de ceux-ci en fonction de leur proximité.

2.1.2 Mesure de proximité sémantique

Nous présentons l'évolution de notre mesure de proximité sémantique chronologiquement en développant les différentes étapes pas à pas. Dans cette section, nous introduisons une mesure largement utilisée dans la prise en compte d'intérêts dans les réseaux pair-à-pair : l'intersection de cache, qui correspond au nombre de fichiers en commun dans le cache de chacun des deux nœuds dont la distance est évaluée.

Avant de poursuivre, force est de remarquer que le contexte de la localité géographique utilisée dans T-Man diffère de celui nécessaire à la localité d'intérêt. En effet, les deux observations suivantes illustrent cette différence :

Symétrie vs. asymétrie Dans le contexte géographique, la mesure de localité d est une mesure de distance : elle respecte les propriétés de

- réflexivité : $\forall x; d(x, x) = 0$
- symétrie : $\forall x, y; d(x, y) = d(y, x)$
- inégalité triangulaire $\forall x, y, z; d(x, z) \leq d(x, y) + d(y, z)$.

Or, dans le contexte sémantique, il est usuel de trouver deux nœuds tel que l'un soit proche sémantiquement de l'autre, mais pour lesquels la réciproque est fautive. Par exemple, prenons un utilisateur α appréciant les musiques de la communauté Rock, Jazz et Java, ainsi qu'un utilisateur β n'étant attiré que par le Jazz. Notons $\xi(a, b)$ la distance sémantique entre a et b par rapport à a . Nous choisissons de normaliser notre mesure (*ie.* $\forall a, b; \xi(a, b) \in [0; 1]$). Il est trivial de conclure : $\xi(\alpha, \beta) > \xi(\beta, \alpha)$. Cette mesure ne possède donc pas la propriété de symétrie et *a fortiori*, d'inégalité triangulaire.

Graphe non-orienté vs. graphe orienté Une conséquence directe de l'observation précédente est l'orientation du graphe de connections du réseau recouvrant. En effet, dans une mesure symétrique, si le nœud α choisit le nœud β comme voisin proche en fonction de cette mesure, β prendra également α pour voisin. Cette propriété n'est plus vraie avec une mesure de distance asymétrique.

Dans le cas d'une mesure de distance symétrique, les arcs du graphe de connections ne sont pas orientés. Dans notre cas, un nœud α peut être voisin d'un nœud β sans pour autant avoir β comme voisin. Nous reviendrons sur cette nécessité dans la partie 2.2.

Dans le reste de ce rapport, la terminologie de *mesure de proximité sémantique* est utilisée. Cette mesure n'étant pas symétrique, la notation $\xi_A(B)$ est adoptée pour représenter la proximité de B par rapport à A .

Intersection de cache

Dans un système de partage de fichiers, chaque nœud propose un ensemble de fichiers aux autres utilisateurs de l'application. Cet ensemble est appelé le *cache* d'un nœud. Afin de capturer la localité d'intérêt, le calcul du cardinal de l'intersection des caches de deux nœuds est une mesure de proximité sémantique très courante [40].

La figure 2.2 illustre l'échelle de valeurs possibles de la mesure de proximité à partir de la relation d'ordre représentée. En effet, plus le nœud B a de fichiers en commun avec le nœud A , plus son profil est proche de celui de A dans le contexte présent, et donc, plus $\xi_A(B)$ est proche de 0.

Malheureusement, cette mesure ne prend pas en compte certaines caractéristiques des systèmes de partage de fichiers pouvant induire des effets de bords tels que la générosité des nœuds et la popularité des fichiers partagés [18].

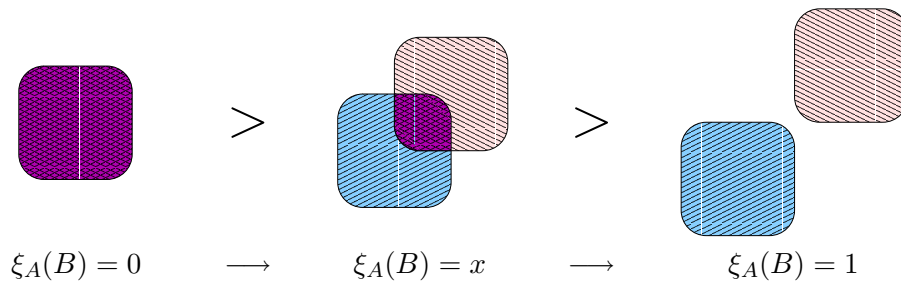


FIG. 2.2 – REPRÉSENTATION DE LA RELATION D’ORDRE SÉMANTIQUE PAR MESURE D’INTERSECTION DES CACHES

Le nœud local A est représenté en bleu par des hachures orientées SO-NE (///) et le nœud distant B est représenté en beige par des hachures orientées NO-SE (\\). L’intersection est représentée par l’addition des deux représentations.

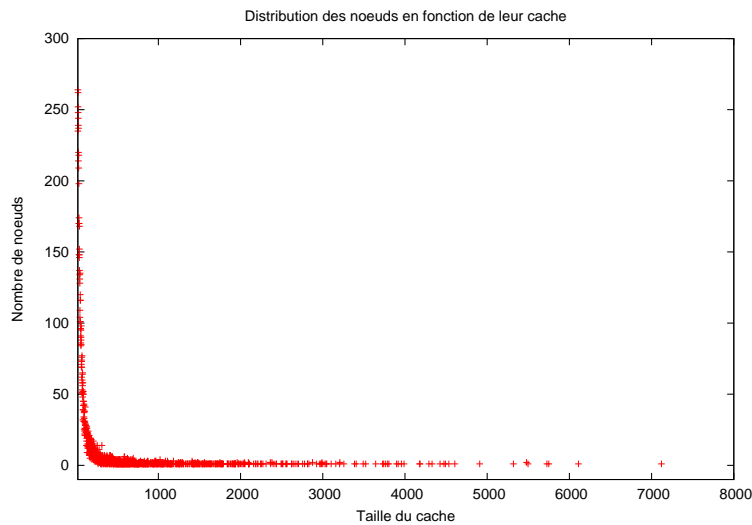


FIG. 2.3 – DISTRIBUTION DES NŒUDS EN FONCTION DE LEUR GÉNÉROSITÉ

Impact de la 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 système. Nous nommons cette quantité la *générosité d’un nœud*. Celle-ci peut biaiser l’évaluation de la distance sémantique [18].

Comme cité plus haut, au sein du projet, nous disposons de plusieurs traces réelles de systèmes de partage de fichiers. Nous porterons notre attention sur une trace du logiciel eDonkey datant d’octobre 2003 et présenté dans le chapitre 3. Grâce à celle-ci, nous avons pu évaluer l’impact des nœuds généreux au sein de ces derniers. La figure 2.3 présente la distribution des nœuds en fonction de leur générosité. Nous pouvons observer la présence d’une quantité non-négligeable de nœuds très généreux. En effet, sur une trace contenant plus de 11 000 nœuds actifs (en dehors des nœuds consommateurs appelés aussi *free-riders*), près de 3 % de ceux-ci proposent au moins 1000 fichiers à la communauté, et plus de 33 % en proposent au moins 100.

Le biais induit par les généreux est le suivant : dans la mesure de proximité s’appuyant sur la taille de l’intersection des caches, la générosité n’est pas prise en compte. Prenons un exemple simple : 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 :

Popularité	1	2	3	4	5	6	7	8	9
Nb de fichiers	934 413	216 271	56 949	24 106	11 672	6 956	4 421	2 837	2 062
10	11	12	13	14	15	16	17	18	19
1 505	1 125	1 008	816	562	457	341	318	266	244

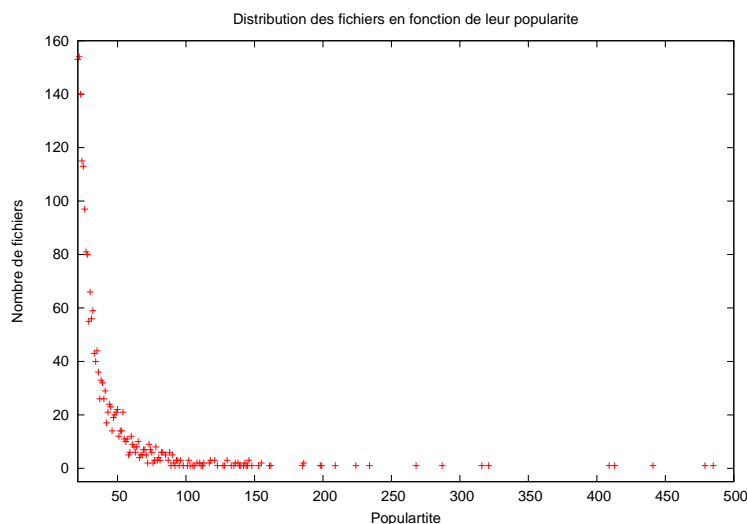


FIG. 2.4 – DISTRIBUTION DES FICHIERS EN FONCTION DE LEUR POPULARITÉ

Le tableau ci-dessus fait correspondre la distribution des fichiers en fonction du nombre de réplicats qu'ils possèdent au sein du système pour les petites valeurs de popularité. Le graphe illustre la distribution des fichiers pour les grandes valeurs de popularité.

posons 10 fichiers en commun par exemple. Dans le contexte défini précédemment, le nœud B a la même probabilité que le nœud C d'être sélectionné comme voisin de A . Or, comme A et C possèdent 66% de leur cache respectif en commun et que pour le nœud B , cela représente seulement 10 % de son cache, il est alors évident que A et C sont plus proches sémantiquement.

De plus, plus un nœud est généreux, plus la probabilité de posséder des fichiers en commun avec n'importe quel autre nœud est grande. Ainsi, un nœud très généreux a une forte probabilité d'être choisi comme voisin sémantique, cela ne reflétant pas une localité d'intérêt authentique mais opportuniste. Surtout, la charge de ce nœud sera nettement plus importante que la moyenne étant donnée cette probabilité. Ainsi, les nœuds généreux risquent d'avoir un degré entrant (nombre de nœuds l'ayant choisi comme voisin) très important et d'être surchargés par les requêtes.

Il s'avère donc nécessaire de prendre en compte la générosité des nœuds afin d'améliorer l'équilibrage de charge du système ainsi que d'affiner la mesure de proximité sémantique.

Impact de la 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 correspond au nombre de réplicats existant dans le système. Comme l'illustre la figure 2.4, la majorité des fichiers possède une popularité inférieure à 10 (réplicats), mais il subsiste un nombre important de fichiers possédant de nombreux réplicats [18].

Une grande popularité induit un effet de bord au même titre qu'une grande générosité. Le biais induit par les fichiers populaires n'influe pas sur l'équilibrage de charge du système mais sur

le grain de la mesure de proximité. En effet, il est intéressant de remarquer que les fichiers rares définissent plus précisément le profil sémantique d'un utilisateur. Deux nœuds ayant un certain nombre de fichiers *rare*s en commun seront plus proches sémantiquement que deux nœuds ayant le même nombre de fichiers *populaires* en commun.

Par exemple, 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. Nous noterons par la suite κ_X le cache du nœud X . Supposons que $\text{Card}(\{f|f \in \kappa_A \cap \kappa_B \wedge f \text{ est rare}\}) = 10$ et que $\text{Card}(\{f|f \in \kappa_A \cap \kappa_C \wedge f \text{ est rare}\}) = 1$, il est trivial de conclure que A est plus proche sémantiquement de B qu'il ne l'est de C .

Un de nos objectifs est donc de prendre en compte la popularité des fichiers dans notre mesure de proximité sémantique, afin d'éviter au maximum les voisins non authentiques.

2.2 Prise en compte des biais

Dans cette partie, nous modélisons la mesure de proximité introduite intuitivement précédemment. Nous posons, pour tout le reste de ce rapport, 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. $\sigma_{A,B} = \kappa_A \cap \kappa_B$;
- τ : le nombre de fichiers populaires présents dans $\sigma_{A,B}$
ie. $\tau = |\{f|f \in \sigma_{A,B} \wedge f \text{ est populaire}\}|$.

Dans la suite, A représente le nœud local, et nous cherchons à évaluer de façon optimale la distance sémantique le séparant d'un nœud B distant.

Par la suite, nous cherchons à maximiser la mesure de proximité $\xi_A(B)$. Plus A et B sont proches sémantiquement, 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 la section 2.1.2 est défini de la manière suivante :

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

2.2.1 Générosité des nœuds

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 κ_A et κ_B sont plus ou moins fournis.

La figure 2.5 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 (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 :

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

Il est possible de modéliser cette mesure de proximité de la manière suivante :

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

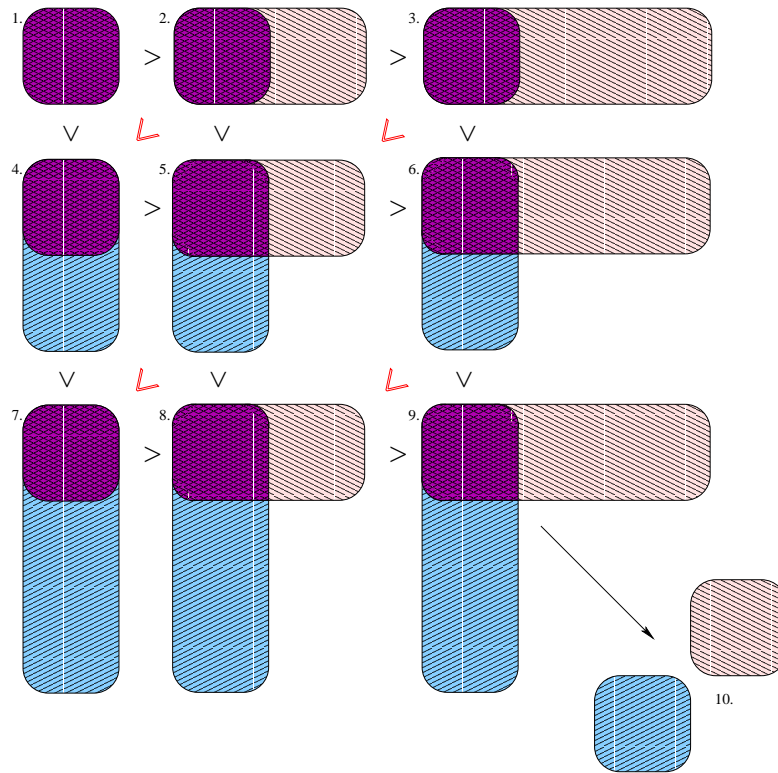


FIG. 2.5 – REPRÉSENTATION DE LA RELATION D'ORDRE SÉMANTIQUE PRENANT EN COMPTE LA GÉNÉROSITÉ DES NŒUDS

Le nœud local A est représenté en bleu par des hachures orientées SO-NE (///) et le nœud distant B est représenté en beige par des hachures orientées NO-SE (\\). L'intersection est représentée par l'addition des deux représentations.

Nous devons à présent choisir un ordre entre les cas transversaux. Par exemple, sur la figure 2.5, quelle est la relation entre le cas 2 et le cas 4? ou entre le cas 3, le cas 5 et le cas 7?

$$\text{Avons-nous : } \begin{cases} \xi_A(B)_{\text{cas 2}} > \xi_A(B)_{\text{cas 4}} \\ \xi_A(B)_{\text{cas 2}} = \xi_A(B)_{\text{cas 4}} ? \\ \xi_A(B)_{\text{cas 2}} < \xi_A(B)_{\text{cas 4}} \end{cases}$$

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 paraît évident que A dans le cas 4 ne pourra obtenir de nouveaux fichiers de la part de B dans l'état actuel des choses. 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 tels que la mesure de proximité prenne en compte l'ordre ainsi fixé (représenté par des \ll rouges sur la figure 2.5). Nous obtenons alors la mesure de proximité suivante :

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

2.2.2 Popularité des fichiers

Comme nous l'avons vu précédemment, cette mesure doit également prendre en compte la popularité des fichiers. Supposons qu'à chaque fichier du système soit associé sa valeur de popularité au sein de celui-ci. Ainsi, en fixant un seuil de popularité à une certaine valeur, nous pouvons déterminer si un fichier est populaire ou non, et le nombre de fichiers populaires présents dans un ensemble de fichiers. Ainsi, nous pouvons déterminer la valeur de τ , introduit dans la section 2.2.

Afin de prendre en compte la popularité des fichiers, nous devons réduire la valeur de $\xi_A(B)$ en fonction de la proportion de fichiers populaires présents au sein de $\sigma_{A,B}$. Pour cela, nous avons fait le choix d'adjointre un facteur multiplicatif à notre mesure afin d'en conserver l'homogénéité. Ce facteur, noté λ , doit suivre les propriétés suivantes :

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

Ainsi, nous obtenons :

$$\lambda = \frac{|\kappa_B| - \tau}{|\kappa_B|} = 1 - \frac{\tau}{|\kappa_B|}$$

Il est également nécessaire de pouvoir modifier le poids de ce facteur multiplicatif sur la mesure de proximité sémantique. Nous choisissons d'y adjoindre également un paramètre de puissance γ qui permet d'optimiser la mesure selon l'heuristique désirée. En effet, la figure 2.6 permet de se rendre compte de l'impact de λ^γ sur la mesure de proximité, en fonction de différentes valeurs de γ .

- 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 :

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

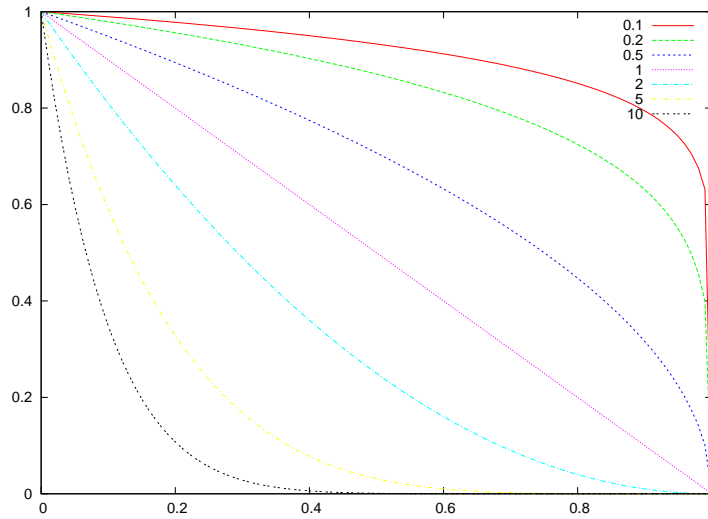


FIG. 2.6 – POIDS DE λ EN FONCTION DE LA VALEUR DE γ

Cette courbe représente l'influence du facteur de puissance γ sur le facteur multiplicatif λ^γ . L'axe des abscisses représente les différentes valeurs de $\frac{\tau}{|\kappa|}$ possible. Les valeurs sont calculées pour $|\kappa| = 50$ et $\gamma \in \{0.1, 0.2, 0.5, 1, 2, 5, 10\}$.

Le chapitre suivant présente la mise en œuvre et l'évaluation de cette mesure de proximité, le banc de tests utilisé pour l'évaluation ainsi que les résultats des expérimentations et une analyse de ceux-ci.

Chapitre 3

Expérimentation

L'expérimentation en environnement réel est très difficile dans le cadre des systèmes distribués à large échelle. En effet, il est nécessaire de disposer d'un nombre de machine suffisamment important pour vérifier l'efficacité dans un environnement à grande échelle. De plus, cette mesure de proximité sémantique entre dans le cadre des systèmes de partage de fichiers, qu'il est d'autant plus difficile d'émuler. Nous choisissons donc de tester notre mesure dans un environnement de simulation de réseau large échelle en utilisant un trace réelle de système de partage de fichiers eDonkey [18].

3.1 Traces disponibles

Pour cela, nous devons simuler un environnement réel de système de partage de fichiers. Nous disposons d'une collection de traces issue du système de partage de fichier eDonkey [1]. Celles-ci sont constituées d'un ensemble de captures de caches sur quelques milliers de nœuds. Ainsi, nous avons un aperçu d'un sous-ensemble du système et de la distribution des différents fichiers mis en partage. Nous pouvons ainsi connaître le degré de générosité de chaque nœud ainsi que la popularité de chaque fichier.

La trace utilisée pour les expérimentations à été capturée en octobre 2003 et analysée dans [19]. Elle contient le cache de 11 291 nœuds partageant un ensemble de 1 268 536 fichiers.

La figure 3.1 est extraite de ces captures. Nous disposons ainsi de nombreux renseignements :

TIME la date de capture du cache ;

ID l'identifiant unique du nœud considéré ;

NFILES le nombre de fichiers composant son cache.

Et pour chaque fichier :

FID l'identifiant unique du fichier ;

FNAME le nom le plus courant du fichier (Plusieurs fichiers identiques peuvent avoir des noms différents, ils sont référencés par un unique identifiant et par le nom de fichier le plus courant) ;

FSIZE la taille du fichier ;

FTAG & FTAGS les caractéristiques du fichier. Le champs FTAG introduit l'information située dans le champs FTAGS suivant.

Malheureusement, nous ne disposons pas de l'information chronologique de la composition des caches. Nous devons donc la simuler au même titre que nous simulons le système. Elle a du reste été utilisé de la sorte dans [18, 40]. La section suivante présente le simulateur utilisé et les choix d'implémentations des structures de données et de la dynamique du système.

```

TIME 317 :02 :00
ID 1
NFILES 159
FID 11679038
FNAME 06 - Nancy Sinatra - These Boots Are Made for Walking.mp3
FSIZE 3883594
FTAG format
FTAGS mp3
FTAG type
FTAGS Audio
FID 10088
FNAME setiathome-3.08.i686-pc-linux-gnu.tar
FSIZE 247808
FTAG format
FTAGS tar

```

FIG. 3.1 – STRUCTURE DE DONNÉE TYPE DE LA TRACE eDONKEY

3.2 Développement d'un simulateur

Afin de tester notre mesure de proximité dans un environnement réaliste, il est nécessaire d'utiliser un simulateur en sus de la trace disponible. De nombreux simulateurs de réseaux existent actuellement. Un certain nombre proposent des simulations de réseaux pair-à-pair à large échelle. Pour chacun d'entre eux, il faut soit fournir une succession d'évènements discrets (impliquant la génération de ceux-ci à partir de la trace), soit apporter des modules additionnels permettant de charger la trace, de créer à partir de celle-ci un réseau recouvrant et d'exécuter un protocole épidémique au sein de celui-ci.

L'investissement en temps et en implémentation s'avère, à peu de choses près, équivalent à l'implémentation d'un simulateur *ad-hoc*. Cette dernière correspond à la solution retenue pour expérimenter notre mesure de proximité sémantique. Ci-dessous, nous présentons succinctement la mise en œuvre de SIMGOSSIP, le simulateur développé durant ce stage.

La figure 3.2 présente le diagramme de classes du simulateur. SIMGOSSIP est implémenté en Java et simule un réseau pair-à-pair de la manière suivante :

La méthode `main()` crée un objet `Network`. Ce dernier contient la liste de tous les nœuds du système ainsi que la liste de tous les fichiers partagés. Au chargement de la trace, pour chaque fichier contenu dans le cache d'un nœud, un lien est créé entre cet objet `File` et les objets `Node` correspondants aux nœuds partageant ce fichier. Puis, nous initialisons le réseau recouvrant arbitrairement. Chaque nœud possède une liste de voisins en plus d'un cache. Il est alors relié à c voisins choisis aléatoirement sur le réseau, c étant la taille de la vue définie en section 1.1.3.

Puis, la procédure de réorganisation épidémique est lancée à l'appel de la méthode `gossip()`. Celle-ci exécute n cycles du protocole, où n est un paramètre du simulateur. À chaque cycle, le processus actif est choisi aléatoirement parmi les participants. À la fin de chaque cycle, le simulateur mesure une estimation du taux de succès d'une recherche de fichier à un saut, c'est-à-dire en consultant uniquement les voisins sémantiques.

Cette estimation est effectuée de la manière suivante : chaque nœud émet une requête à chacun de ses voisins concernant un fichier rare choisi aléatoirement dans son cache. Si au moins un de ses voisins *directs* le possède dans son cache, alors le taux de succès est incrémenté. Puis, cette valeur est normalisée par le nombre de nœuds présents dans le système. Cette méthode bien que peu réaliste (un utilisateur ne va pas émettre une requête concernant un fichier qu'il possède

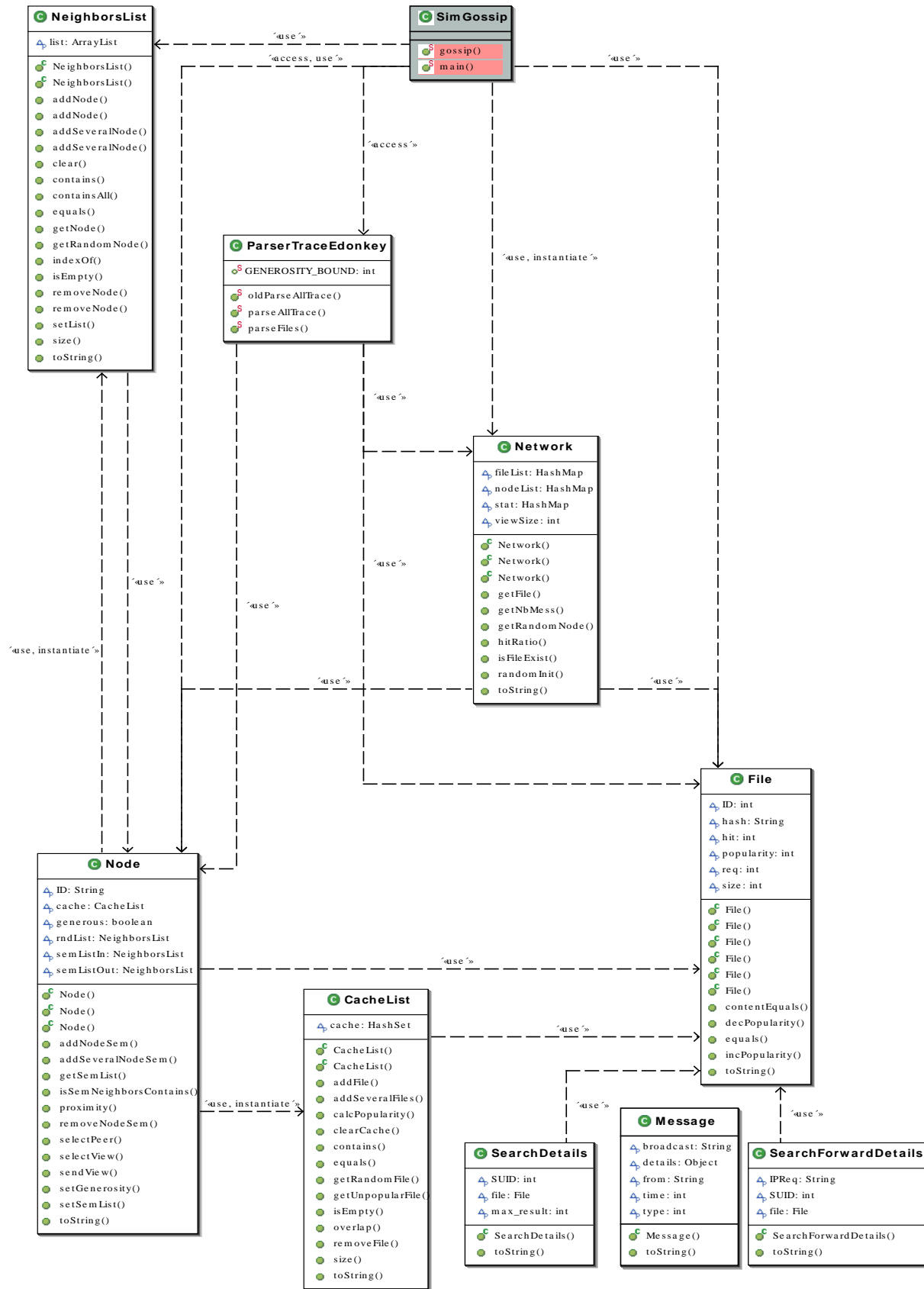


FIG. 3.2 – DIAGRAMME DE CLASSE DU SIMULATEUR SIMGOSSIP

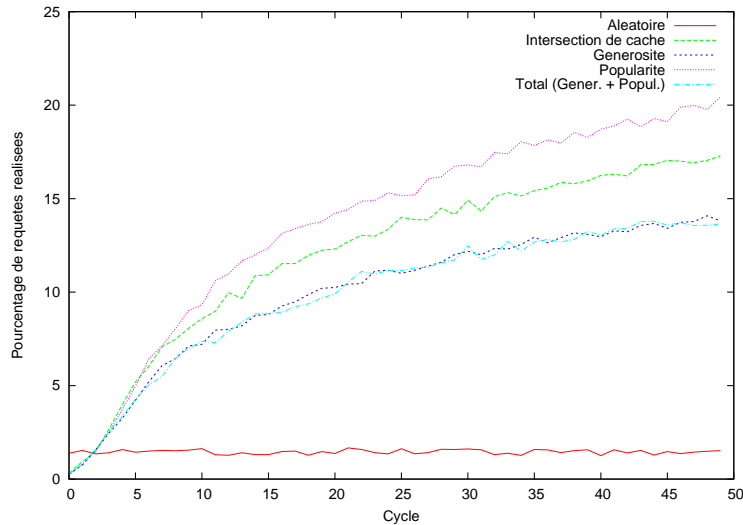


FIG. 3.3 – TAUX DE SUCCÈS À UN SAUT EN FONCTION DE L'AVANCEMENT DE LA PROCÉDURE ÉPIDÉMIQUE

déjà), reflèterai tout de même correctement le profil sémantique d'un utilisateur et permet de comparer avec l'existant [40].

La partie suivante présente les résultats obtenus avec de nombreuses simulations et fournit une évaluation de ceux-ci en fonction des différents points de vue cités dans le chapitre 2.

3.3 Évaluation

Afin de correctement illustrer chacun des points étudiés ci-avant dans la recherche d'une mesure de proximité sémantique, nous présentons tout d'abord des résultats globaux, avant de préciser pour chacun des biais, l'apport de notre mesure.

Afin d'homogénéiser les résultats, le simulateur est paramétré identiquement quelque soit la simulation :

- Taille de la vue : $c = 20$.
- Nombre de cycle : $n = 50$.
- Limite de générosité : 300 fichiers ;
Les nœuds possédant plus de 300 fichiers seront considérés comme généreux par le système.
- Limite de popularité : 10 réplicats ;
Les fichiers possédant plus de 10 réplicats seront considérés comme populaires par le système.
- Nombre de fichiers minimums dans le cache : 20 ;
Ce choix permet de ne pas prendre en compte les nœuds consommateurs ou *free-riders*.

3.3.1 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 directs).

La figure 3.3 présente les résultats obtenus pour chacune des mesure de proximité suivante :

- Aléatoire (*ie.* non sémantique) *en rouge*
- Intersection de cache *en vert*

- $\xi_A(B)$ avec $\alpha = 1/3$, $\beta = 2/3$ et $\gamma = 0$ (inhibe la prise en compte de la popularité) *en bleu marine*
- $\xi_A(B)$ avec $\alpha = 1$, $\beta = 0$ et $\gamma = 1$ (inhibe la prise en compte de la générosité) *en rose*
- $\xi_A(B)$ avec $\alpha = 1/3$, $\beta = 2/3$ et $\gamma = 1$ *en bleu ciel*

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 le calcul du cardinal de l'intersection des caches.

Au sein des sections suivantes, nous revenons sur chacun des effets de bords introduits dans le chapitre 2. Nous analysons la figure 3.3 pour chacun d'entre eux afin d'explicitier les différents comportements du système.

3.3.2 Prise en compte de la générosité

Nous présentons ici les conséquences de la prise en compte de la générosité dans la mesure de proximité. 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.

Taux de succès

Nous observons sur la figure 3.3 une diminution non négligeable du taux de succès dans le cas titré « Générosité ». Cette diminution s'explique logiquement. En effet, 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 donnée que les sources riches, *ie.* les nœuds possédant de très nombreux fichiers, seront moins souvent choisis comme voisins sémantiques et que la mesure du taux de succès s'effectue à un saut.

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

Équilibrage de charge

La figure 3.4 illustre la répartition de la charge entrante des nœuds du système *ie.* le nombre de nœuds l'ayant choisi comme voisin sémantique. Cette courbe possède la même allure quelle que soit la valeur de α , β et γ .

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, la figure 3.5 permet de comparer l'avantage de la prise en compte de la générosité des nœuds par rapport à la méthode d'intersection des caches. Les valeurs représentées sur cette courbe sont cumulatives. Nous avons choisi ce mode de représentation car il permet de mieux observer les points de grands degrés. Nous pouvons remarquer que les deux courbes représentées possèdent effectivement la même allure, mais le nœud le plus chargé avec la méthode d'intersection des caches possède un degré presque deux fois plus élevé que celui avec la méthode prenant en compte la générosité (1702 liens contre 990 dans notre méthode).

Nous pouvons à présent discuter de l'impact de α et β sur l'équilibrage de charge.

Influence de α et β

La figure 3.6 représente la même courbe que la figure 3.5 mais pour des valeurs de α et β différentes.

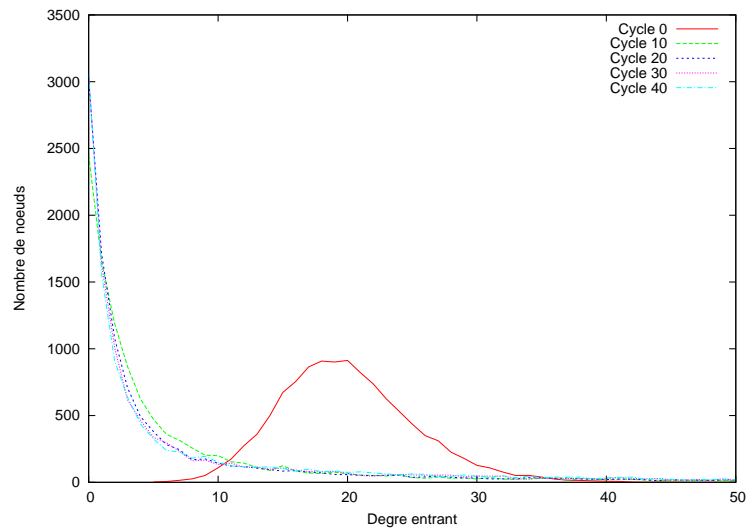


FIG. 3.4 – DISTRIBUTION DES NŒUDS DU SYSTÈME EN FONCTION DE LA TAILLE DE LEUR DEGRÉ ENTRANT ET DE L'AVANCEMENT DE LA PROCÉDURE

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

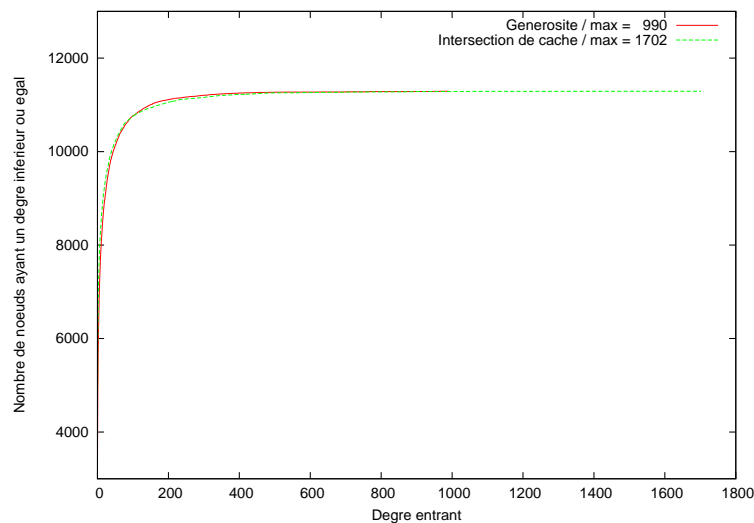


FIG. 3.5 – DISTRIBUTION CUMULATIVE (CDF) POUR L'INTERSECTION DE CACHE ET LA GÉNÉROSITÉ

Cette courbe présente la distribution des degrés entrant du graphe de connections du réseau recouvrant de manière cumulative. Les valeurs correspondantes aux degrés ne sont pas égales au nombre de nœuds possédant ce degré entrant mais à la somme des nœuds possédant un degré entrant inférieur ou égal à celui-ci.

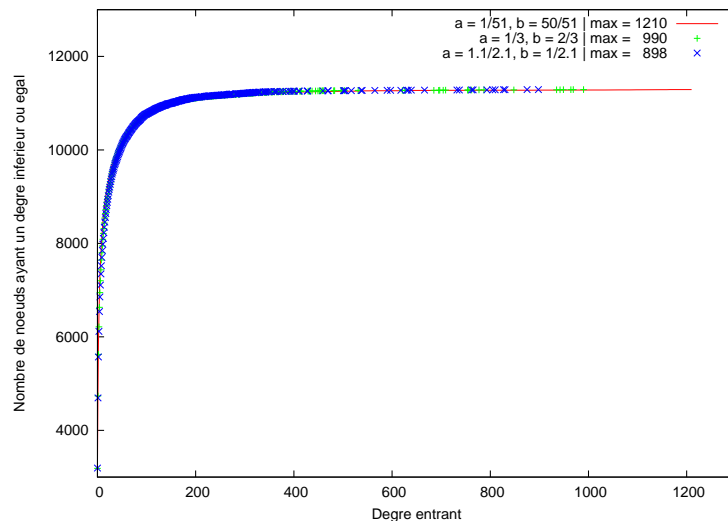


FIG. 3.6 – INFLUENCE DE α ET β SUR LA DISTRIBUTION CUMULATIVE

Cette courbe présente la distribution des degrés entrant du graphe de connections du réseau recouvrant de manière cumulative. Les valeurs correspondantes aux degrés ne sont pas égales au nombre de nœuds possédant ce degré entrant mais à la somme des nœuds possédant un degré entrant inférieur ou égal à celui-ci.

Nous pouvons observer que plus α et β sont proches, plus l'équilibrage de charge des nœuds s'améliore. L'impact de ces derniers sur le taux de succès moyen n'est pas flagrant. Cependant, nous ne représentons pas la courbe comparative mais préférons plutôt résumer les valeurs importantes. Ci-dessous, le tableau récapitulatif pour différentes valeurs de α et β :

α	β	Degré maximum	Taux de succès moyen au cycle 50
$\frac{1}{2.1}$	$\frac{1.1}{2.1}$	898	14,53%
$\frac{1}{3}$	$\frac{2}{3}$	990	13,80%
$\frac{1}{51}$	$\frac{50}{51}$	1210	12,23%

3.3.3 Prise en compte de la popularité

Nous présentons ici 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 les effets du membre de ξ prenant en compte la générosité des fichiers.

Taux de succès

Nous pouvons observer sur la figure 3.3 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 un meilleur taux de succès.

Taux de succès en fonction de la popularité des fichiers

La courbe 3.7 permet d'évaluer précisément le taux de succès en fonction de la popularité.

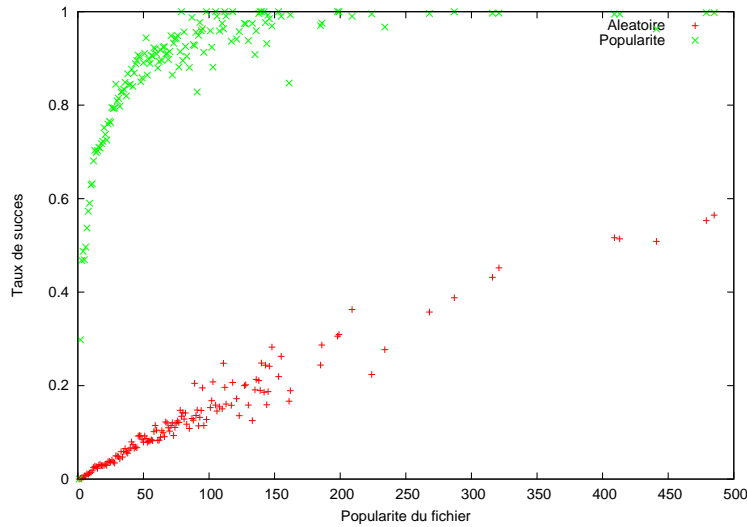


FIG. 3.7 – TAUX DE SUCCÈS EN FONCTION DE LA POPULARITÉ DES FICHIERS

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

Cette courbe est issue d'un test effectué à la fin de la procédure 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 3.7 qui représente le taux de succès en fonction de la popularité des fichiers.

Cette courbe 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 réplicats sur le réseau), le taux de succès atteint 80 % contre 4 % avec la méthode aléatoire.

La figure 3.8 illustre la différence du taux de succès sur les fichiers rares (moins de 20 réplicats) entre la méthode d'intersection de caches et notre mesure 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. En effet, la méthode prenant en compte la popularité des fichiers permet une amélioration comprise entre 1,5 et 7 % pour les tests effectués.

Influence de γ

La valeur du paramètre γ influe peu sur le taux de succès moyen et l'équilibrage de charge tant qu'il reste supérieur à $1/10$. Pour des valeurs de $\gamma \in [0; 0, 1]$, le système tend vers la méthode d'intersection des caches.

Le tableau récapitulatif ci-dessous illustre les influences limitées du paramètre γ :

γ	Degré maximum	Taux de succès moyen au cycle 50
1/3	1200	20,82%
1	1244	20,44%
3	1228	20,32%
10	1214	20,55%

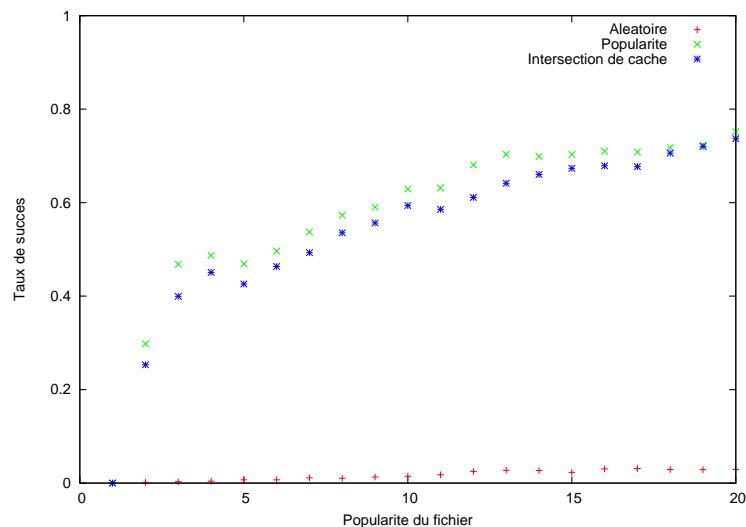


FIG. 3.8 – TAUX DE SUCCÈS EN FONCTION DE LA RARETÉ DES FICHIERS

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.

3.3.4 Évaluation de $\xi_A(B)$

Nous pouvons observer que lorsque la générosité des nœuds et la popularité des fichiers sont pris en compte séparément, le biais associé s'en retrouve diminué.

Dans les expériences menées au cours de ce stage, nous avons cherché à intégrer la popularité et la générosité. 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 perte d'efficacité du taux de succès induit par la prise en compte de la générosité, dans le cadre des traces disponibles. Comme l'illustre la figure 3.3, la courbe « Total » côtoie la courbe « Générosité » sans être supérieure. Combiner efficacement ces deux approches représente une perspective à ce travail.

Conclusion et perspectives

Ce rapport présente les travaux que j'ai effectués dans le cadre de mon stage de Master de Recherche en Informatique sous la direction d'Anne-Marie Kermarrec, au sein du projet PARIS à l'IRISA durant la période de février à juin 2005. Nous avons introduit celui-ci par un aperçu élargi du domaine de recherche dans lequel ce stage s'est déroulé. Puis, nous avons présenté mes contributions, notamment la conception d'une mesure de proximité sémantique permettant de prendre en compte deux biais principaux de la mesure usitée jusqu'à présent par la communauté. Nous avons conclu ce rapport par une présentation succincte de la mise en œuvre d'un simulateur de réseau pair-à-pair large échelle et une analyse des différents résultats issus des expérimentations effectuées sur une trace du logiciel eDonkey datant d'octobre 2003.

Nous avons observé que la prise en compte d'une proximité sémantique améliorait notablement le taux de succès des recherches dans un système pair-à-pair de partage de fichiers. La prise en compte de la générosité des nœuds permet d'obtenir un meilleur équilibrage de charge, mais diminue ostensiblement le taux de succès des requêtes sur les voisins immédiats. La prise en compte de la popularité des fichiers partagés au sein de l'application permet au contraire d'augmenter de manière non négligeable ce taux de succès, adjoint d'un équilibrage de charge très acceptable.

Malheureusement, l'espoir de compenser la diminution du taux de succès de la prise en compte de la générosité des nœuds en intégrant les deux limitations des biais dans la mesure de proximité sémantique n'est pas encore réalisé. En effet, dans les expériences menées au cours de ce stage, la prise en compte de la popularité n'arrive pas à compenser la perte d'efficacité du taux de succès induit par la prise en compte de la générosité.

Plusieurs perspectives de travaux de recherches sont en cours. D'une part, la prise en compte de la popularité analysée dans la section 3.3.3 s'est effectuée de manière omnisciente. Or, la popularité réelle d'un fichier dans un système de partage de fichiers est une information globale. Il faut connaître la localisation de tous les réplicats afin de quantifier cette valeur. Or, dans un système pair-à-pair, un des facteurs clé au passage à l'échelle est la vision uniquement locale des nœuds. Un problème encore ouvert est donc de déterminer une approximation de la popularité d'un fichier de manière locale. Plusieurs pistes sont déjà en cours d'évaluation.

De plus, durant mon stage, nous avons engagé un partenariat international avec l'université libre d'Amsterdam (*Vrije Universiteit*). Nous collaborons avec Elizabeth Ogston afin d'étudier l'impact du groupement de nœuds (*clustering*) sémantique dans les systèmes de partage de fichiers et l'efficacité de notre mesure de proximité sémantique au sein de leur algorithme de création et de détection de groupements.

Enfin, j'ai commencé à travailler avec Étienne Rivière sur une mesure de pertinence des mots-clés dans les requêtes de ces systèmes. L'observation des traces disponibles met en évidence une différence entre le profil sémantique reflété par l'étude du cache d'un nœud et celui reflété par l'étude des requêtes vers les autres nœuds. De manière générale, ce comportement est assez difficile à prendre en compte directement dans la mesure de proximité. Il pourra donc faire l'objet de prochaines études.

Annexe A

Glossaire

Ci-dessous, un récapitulatif non-exhaustif des termes anglophones usités dans la communauté ainsi que leur traduction utilisée dans ce rapport.

Average path length Longueur moyenne d'un chemin

Bootstrapping Amorçage

Clustering Groupement de nœuds

Clustering coefficient Coefficient de groupement de nœuds

Distributed Hash Table Table de hachage distribuée

Fault-tolerant Tolérance aux fautes

Free-riders Nœuds consommateurs

Gossip-based protocol Protocole de diffusion épidémique

Last Recent Used Utilisé le plus récemment

Membership Appartenance

Overlay network Réseau recouvrant

Pattern analysis Analyse de modèle

Pattern discovery Découverte de modèle

Peer-to-peer Pair-à-pair

Pull Demande d'informations à un nœud

Push Envoi d'informations à un nœud

Push/Pull Échange d'informations avec un nœud

Random rewiring procedure Procédure aléatoire de réorganisation des liens entre les nœuds

Rolling-index Index rotatif

Self-organising Auto-organisation

Semantic Overlay Network Réseau recouvrant sémantique

Semantic Small World Réseau « Petit monde » sémantique

Web Usage Mining Exploitation de l'utilisation du WWW

Bibliographie

- [1] The eDonkey 2000 project. <http://www.edonkey2000.com/>.
- [2] The eMule project. <http://www.emule-project.net/>.
- [3] The Gnutella project. <http://www.gnutella.com/>.
- [4] The KaZaA project. <http://www.kazaa.com/>.
- [5] The Napster project. <http://www.napster.com/>.
- [6] The Planet-Lab project. <http://www.planet-lab.org/>.
- [7] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science Reports*, Vol. 286 :509–512, Oct. 1999. www.sciencemag.org/cgi/content/abstract/286/5439/509.
- [8] J. Broekstra, M. Ehrig, P. Haase, F. van Harmelen, A. Kampman, M. Sabou, R. Siebes, S. Staab, H. Stuckenschmidt, and C. Tempich. A metadata model for semantics-based peer-to-peer systems. In *The P2P Workshop at the Twelfth International World Wide Web Conference (P2PW - WWW'03)*, Budapest, Hungary, May 2003. <http://www.cs.vu.nl/~frankh/postscript/{P2P}@WWW03.pdf>.
- [9] Y. Busnel. Prise en compte d'aspects sémantiques dans la construction d'un réseau pair-à-pair. In *The 3rd Manifestation des Jeunes Chercheurs dans le domaine des STIC (Majec'STIC'05)*, Rennes, France, Nov. 2005.
- [10] M. Castro, M. Costa, and A. Rowstron. Should we build Gnutella on a structured overlay? In *The 2nd Workshop on Hot Topics in Networks (HotNets-II)*, MIT, Cambridge, MA, USA, Nov. 2003. <http://research.microsoft.com/~antr/MS/Structella-HotNets.pdf>.
- [11] M. Castro, P. Druschel, Y. C. Hu, and A. Rowstron. Topology-aware routing in structured peer-to-peer overlay networks. Technical Report MSR-TR-2002-82, Microsoft Research, Cambridge, UK, Sept. 2002. <ftp://ftp.research.microsoft.com/pub/tr/tr-2002-82.pdf>.
- [12] 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, Sept. 2003. <ftp://ftp.research.microsoft.com/pub/tr/tr-2003-52.ps>.
- [13] A. Crespo and H. Garcia-Molina. Semantic overlay networks for P2P systems. Technical report, Database Group, Stanford University, Stanford, CA, USA, Sept. 2002. <http://www-db.stanford.edu/~crespo/publications/oP2P.pdf>.
- [14] C. Delannoy. *Programmer en Java*. Eyrolles, deuxième édition mise à jour edition, May 2002.
- [15] P. T. Eugster, R. Guerraoui, S. B. Handurukande, A.-M. Kermarrec, and P. Kouznetsov. Lightweight probabilistic broadcast. *ACM Transactions on Computer Systems (TOCS)*, Vol. 21(4) :341–374, Nov. 2003. <http://www.irisa.fr/paris/Biblio/Papers/Kermarrec/EugHanGueKerKou03ACMT0CS.pdf>.

- [16] A. J. Ganesh, A.-M. Kermarrec, and L. Massoulié. Scamp : Peer-to-peer lightweight membership service for large-scale group communication. In *The 3rd International Workshop on Networked Group Communications (NGC 2001)*, London, UK, Nov. 2001. <http://www.irisa.fr/paris/Biblio/Papers/Kermarrec/GanKerMas01NGC.pdf>.
- [17] 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, Feb. 2004. <http://iptps04.cs.ucsd.edu/papers/le-fessant-clustering.pdf>.
- [18] 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, Sept. 2004. <http://pauillac.inria.fr/~lefessan/papers/sigops2004.pdf>.
- [19] 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. Technical Report Publication interne n 1697, Institut de Recherche en Informatique et Systèmes Aléatoires, Rennes, France, Feb. 2005.
- [20] 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, May 2004. <http://www.cs.unibo.it/techreports/2004/2004-07.pdf>.
- [21] 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, Oct. 2004. <http://www.cs.unibo.it/bison/publications/middleware2004.pdf>.
- [22] A.-M. Kermarrec. Diffusion fiable large-échelle. Technical Report Documents d'habilitation à diriger les recherches, IRISA + IFSIC, Rennes, France, Dec. 2002.
- [23] J. Kleinberg. The small-world phenomenon : An algorithmic perspective. Technical Report CCS-TR-99-1776, Department of Computer Science, Cornell University, Ithaca, NY, USA, Oct. 1999. <http://www.cs.cornell.edu/home/kleinber/swn.ps>.
- [24] L. Lamport. *L^AT_EX 2_ε, a Document Preparation System*. Addison-Wesley publishing company, second edition edition, Aug. 1999.
- [25] M. Li, W.-C. Lee, and A. Sivasubramaniam. Semantic small world : An overlay network for peer-to-peer search. In *The 12th IEEE International Conference on Network Protocols (ICNP'04)*, Berlin, Germany, Oct. 2004. <http://www.ieee-icnp.org/2004/papers/6-2.pdf>.
- [26] B. T. Loo, R. Huebsch, I. Stoica, , and J. M. Hellerstein. The case for a hybrid P2P search infrastructure. In *The 3rd International Workshop on Peer-to-Peer Systems (IPTPS'04)*, San Diego, CA, USA, Feb. 2004. http://www.cs.berkeley.edu/~boonloo/papers/gnutella_iptps.pdf.
- [27] S. Marti, P. Ganesan, and H. Garcia-Molina. DHT routing using social links. In *The 3rd International Workshop on Peer-to-Peer Systems (IPTPS'04)*, San Diego, CA, USA, Feb. 2004. <http://www.stanford.edu/~smartipapers/2004-4.pdf>.
- [28] L. Massoulié, A.-M. Kermarrec, and A. J. Ganesh. Network awareness and failure resilience in self-organising overlay networks. In *The 22nd Symposium on Reliable Distributed Systems (SRDS'03)*, Florence, Italy, Oct. 2003. <http://research.microsoft.com/research/network/talks/srds03.ppt>.

- [29] S. Milgram. The small world problem. In *Psychology Today*, 1967.
- [30] A. Montresor, M. Jelasity, and O. Babaoglu. Robust aggregation protocols for large-scale overlay networks. Technical Report UBLCS-2003-16, University of Bologna, Department of Computer Science, Bologna, Italy, Dec. 2003. <http://www.cs.unibo.it/techreports/2003/2003-16.pdf>.
- [31] W. Nejdl, W. Siberski, U. Thaden, and W.-T. Balke. Top- k query evaluation for schema-based peer-to-peer networks. In *3rd International Semantic Web Conference (ISWC'04)*, Hiroshima, Japan, Nov. 2004. http://www.kbs.uni-hannover.de/Arbeiten/Publikationen/2004/topk_iswc.pdf.
- [32] 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, Aug. 2001. <http://www.acm.org/sigs/sigcomm/sigcomm2001/p13-ratnasamy.pdf>.
- [33] E. Rivière and P. Gauron. Rechercher parmi ses pairs ou quand le hasard ne fait pas si bien les choses. In *The 3rd Manifestation des Jeunes Chercheurs dans le domaine des STIC (Majec'STIC'05)*, Rennes, France, Nov. 2005.
- [34] 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, Nov. 2001. <http://research.microsoft.com/~antr/PAST/pastry.pdf>.
- [35] 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, Apr. 2003. http://www.ieee-infocom.org/2003/papers/53_01.PDF.
- [36] J. Srivastava, R. Cooley, M. Deshpande, and P.-N. Tan. Web Usage Mining : Discovery and applications of usage patterns from web data. In *The Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2000)*, Boston, MA, USA, Aug. 2000. <http://www.acm.org/sigkdd/explorations/issue1-2/srivastava.pdf>.
- [37] 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, Aug. 2001. http://www.pdos.lcs.mit.edu/papers/chord:sigcomm01/chord_sigcomm.pdf.
- [38] C. Tang, Z. Xu, and S. Dwarkadas. Peer-to-peer information retrieval using self-organizing semantic overlay networks. Technical report, HP Laboratories, Palo Alto, CA, USA, Nov. 2002. <http://www.eecs.harvard.edu/~mema/courses/cs264/papers/ir-sigcomm2003.pdf>.
- [39] S. Voulgaris, A.-M. Kermarrec, L. Massoulié, and M. van 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, May 2004. <http://www.cs.vu.nl/~spyros/papers/ftdcs.04.pdf>.
- [40] S. Voulgaris and M. van Steen. Epidemic-style management of semantic overlays for content-based searching. In *Euro-Par'05*, Lisboa, Portugal, Aug. 2005.
- [41] H. J. Wang, Y.-C. Hu, C. Yuan, Z. Zhang, and Y.-M. Wang. Friends troubleshooting network : Towards privacy-preserving, automatic troubleshooting. In *The 3rd International Workshop on Peer-to-Peer Systems (IPTPS'04)*, San Diego, CA, USA, Feb. 2004. <http://iptps04.cs.ucsd.edu/papers/wang-ftn.pdf>.

-
- [42] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Letters to Nature*, Vol. 393 :440–442, June 1998. http://tam.cornell.edu/SS_nature_smallworld.pdf.
- [43] B. Y. Zhao, J. Kubiawicz, and A. D. Joseph. Tapestry : An infrastructure for fault-tolerant wide-area location and routing. Technical Report UCB/CSD-01-1141, Computer Science Division (EECS), University of California, Berkeley, CA, USA, Apr. 2001. <http://www.cs.berkeley.edu/~ravenben/publications/CSD-01-1141.pdf>.