HAL
open science

Autour du pair-à-pair : distribution de contenus, réseaux à préférences acycliques

Fabien Mathieu

To cite this version:

Fabien Mathieu. Autour du pair-à-pair : distribution de contenus, réseaux à préférences acycliques. Réseaux et télécommunications [cs.NI]. Université Pierre et Marie Curie - Paris VI, 2009. tel-00667414

HAL Id: tel-00667414

https://theses.hal.science/tel-00667414v1

Submitted on 7 Feb 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Autour du pair-à-pair

distribution de contenus, réseaux à préférences acycliques

Fabien Mathieu

Habilitation à Diriger des Recherches

Université Pierre et Marie Curie (UPMC – Paris 6)

Soutenance le 11 février 2009 devant le jury composé de :

Pascal Felber, Professeur, Université de Neuchâtel	Rapporteur
Pierre Fraigniaud, DR CNRS	Rapporteur
Jérôme Galtier, Orange Labs	Examinateur
Laurent Massoulié, Thomson Technology Paris Laboratory	Examinateur
Philippe Robert, DR INRIA	Rapporteur
Sébastien Tixeuil, Professeur, Université Pierre et Marie Curie	Examinateur
Laurent Viennot, CR INRIA	Examinateur

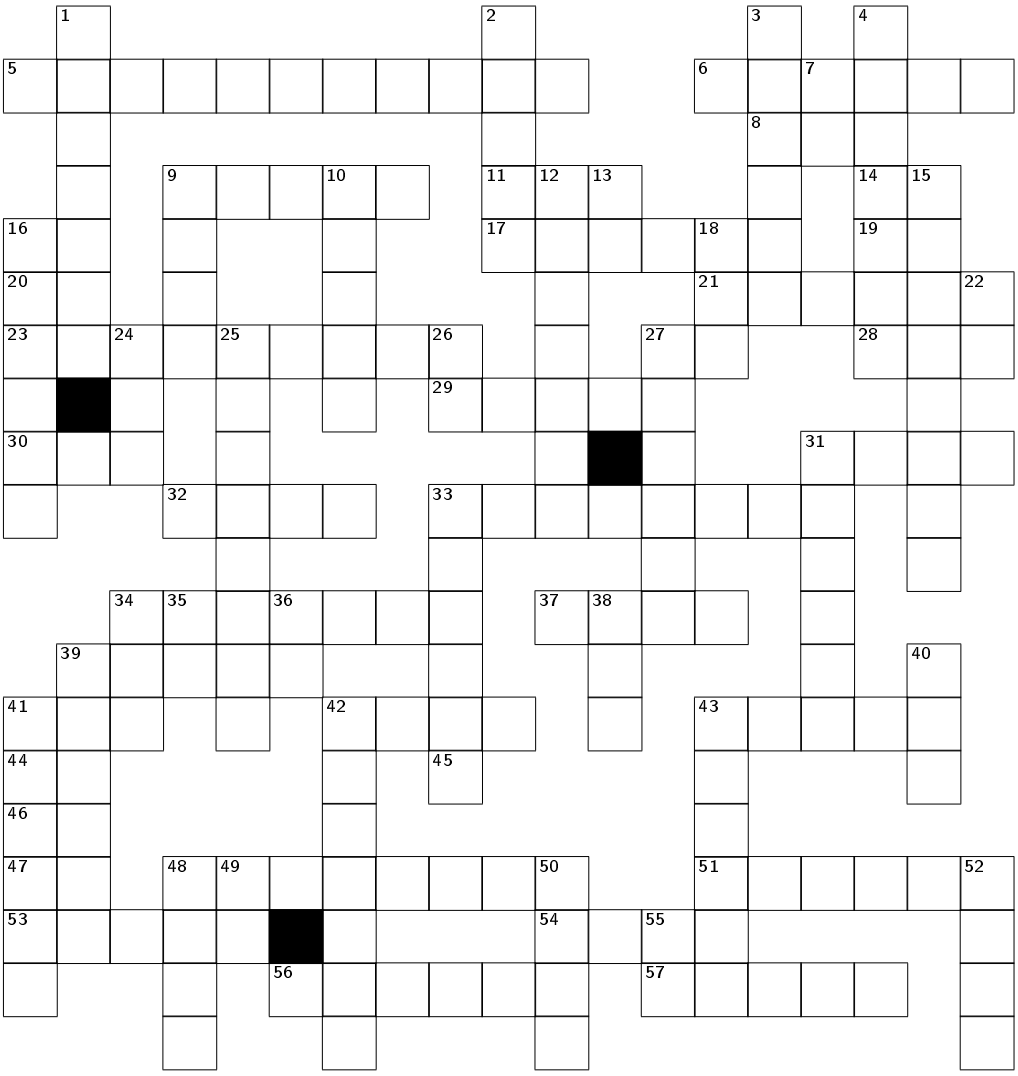
All peers are equal, but some peers are more equal than others.
George Orwell, *the server farm*

V2.1

Remerciements

Les remerciements sont certainement la partie d'un mémoire la plus délicate à écrire. Comme mon père était un grand cruciverbiste, j'ai décidé de lui rendre cet hommage, transférant au passage la difficulté de l'auteur au lecteur. Quelques petites précisions utiles afin d'attaquer ces remerciements dans les meilleures conditions :

- Beaucoup de définitions font appel à des anecdotes personnelles. Il est donc normal de ne pas pouvoir résoudre toute la grille (en cela, ces remerciements sont somme toute comme les autres).
- J'espère cependant que chacun retrouvera ses cases. Dans le cas contraire, $V31$.
- Pour les organismes vivants, je ne donne que le prénom ou l'espèce. En cas de doublé, une seule entrée sur la grille, mais éventuellement deux définitions.
- Certains mots de la grille ne sont pas des remerciements, mais sont juste là pour diminuer la difficulté (ou l'augmenter selon certains). Ce sont souvent (mais pas toujours) les plus courts.



Horizontalement 5 C'est vilain de le faire, mais je suis content qu'ils aient accepté 6 Où l'on chasse les têtes 8 Grosse souris sans fil 9 Annexe parisienne de Montpellier 11 Total 14 Pas Dell 16 Où l'on inverse la notation 17 Atos (l'autre) 19 Système d'autoroutes aux USA 20 Prénom balafré 21 Zen 23 Expert démoniaque 27 Messe du soir 28 Rue des grosses têtes 29 Calamité 30 Melmacien 31 S'est fait avoir 32 Honorable 33 Blonde insulaire 34 Fumeur de Drums 37 Association criminelle se réunissant le MARDI 39 Fruit indigeste pour les pandas 41 C'est le meilleur qui s'en va en premier 42 Là quand on a besoin d'eux 43 Apprentice 44 Fin de soirée 45 Avec ça, on gagne bien sa vie 46 Au cœur de l'horreur 47 Rêve de chercheur 48 Amateurs de plaisir solitaire 51 Il faut le voir pour le croire 53 Jase 54 Faire fait 56 Une Pomme tombée sur ma tête (merci d'être velu) 57 Canadienne sans fil

Verticalement 1 C'est compliqué 2 Motif de plainte 3 Ma hyène 4 Un homme à marier ? 7 Sous-sol 9 Made in Italia 10 What else ? 12 Push and Beyond – Master 13 Quinze fois rien 15 Grand prénom du rap – Amateur de péplums 16 Le petit se fait rouler (mais pas lui !) 18 6 heures du mat' les nuits de deadline 22 Mesure un moment 24 Tombeau présidentiel 25 Muridé informaticien (qu'il repose en paix sur son petit nuage de gruyère) 26 Aux normes 27 D'artagnan 31 À ceux qui recherchent vainement leur nom ici (cette marque d'affection me va droit au cœur) 33 Le monde de Neo 34 Ce n'est pas la taille qui compte 35 Cocotte 36 Un bleu qui n'a pas lu 38 Baba cool 39 Routeur glouton 40 Expert de l'égalité 41 Master of the WebWorld 42 Aramis 43 White Russian 48 Vue imprenable sur la Tour Eiffel 49 Additif 50 Liquide mis en banque 52 Sans hache, j'écris ton nom 55 À fond de train

Avant-propos

La scène se passe dans un débit de boissons, par une longue après-midi d'un mois d'été finissant. Deux amis se prélassent à la terrasse. La bière est fraîche, et la discussion légère. L'objet du débat : l'existence de l'âme sœur et la mesure quantitative de l'amour. Pour l'un des deux protagonistes, l'espace des sentiments est une géométrie extrêmement complexe qui n'admet pas d'extremum absolu. L'adultère n'est alors rien de plus que l'application amoureuse de la méthode d'optimisation dite du *recuit simulé*. De fil en aiguille, le sujet dévie sur un article de *science et vie* qui traite d'un modèle mathématique qui donne, entre autres, une vision très croustillante des relations hommes-femmes.

C'est ainsi que je fis connaissance, en août 2005, avec la théorie des mariages stables. Et que je trouvai sans le savoir mon principal projet de recherche des trois années à venir, qui finirait par motiver la présente habilitation.

Cela m'amène à remarquer l'analogie entre la recherche et le ski : sur un domaine connu et fréquenté, il y a peu de risques de se perdre. Le terrain est généralement (plus ou moins) bien balisé, et il n'est pas très difficile de savoir quelles sont les pistes intéressantes à descendre.

À l'inverse, un domaine inconnu procure la joie de foncer dans la poudreuse, là où la main du chercheur n'a jamais mis le pied. Mais il faut bien admettre que l'on ne sait en général pas trop où l'on va.

J'ai ainsi vu des théorèmes d'une puissance extraordinaire se laisser démontrer en quelques minutes, tandis que d'autres ont résisté (et résistent toujours), nuit blanche après nuit blanche, à tous les assauts subis. Qui n'a jamais rêvé de ces graphes de terrain, de ces pairs lointains peuplés de légendes ou d'une richesse soudaine qui se conquerrait au détour d'un chemin d'une chaîne de Kieschnick ? Qui n'a jamais souhaité voir des équations non locales guider ses pas, au cœur d'une configuration stable, vers la richesse et l'histoire des mystérieuses distributions acycliques ?

Mais je m'éloigne déjà de l'avant-propos pour rentrer dans le vif du sujet. J'espère que le lecteur prendra autant de plaisir à lire ce mémoire que j'en ai pris à l'écrire. Attention cependant, si l'introduction a été écrite de manière à être accessible à un public le plus large possible, il est possible de rencontrer au gré des chapitres quelques formules et idées de preuves. En particulier au cours du chapitre 4.

Table des matières

1 Introduction **1**
1.1 Histoire subjective du pair-à-pair 1
1.2 Définir le pair-à-pair 2
1.3 Thématiques de recherche 4

2 Positionnement **7**

3 Distribution de contenu **9**
3.1 Distribuer 9
3.2 Dimensionner 10
3.3 Simplifier 11
3.4 Décentraliser 12

4 Réseaux à préférences acycliques **13**
4.1 Origine 13
4.2 Les bases 14
4.3 Caractérisation de l'auto-stabilisation 21
4.4 Les configurations stables 26
4.5 Conclusion 36

5 Synthèse et perspectives **37**

Chapitre 1

Introduction

Ce mémoire présente une partie majeure des travaux de recherche que j'ai menés depuis ma thèse de doctorat, soutenue en décembre 2004. Il est centré autour du pair-à-pair (P2P), avec une attention toute spéciale dédiée au thème des réseaux à préférences. Mes travaux situés hors du champ P2P, principalement des travaux autour du PageRank de Google, ne sont pas développés (les références sont disponibles dans le Curriculum Vitæ joint à ce mémoire). Les deux principales raisons de ce choix éditorial sont la volonté de présenter un ensemble articulé autour d'une problématique commune (le P2P), ainsi que l'intéressante contrainte posée par la rédaction d'un mémoire de taille limitée.

Dans ce chapitre d'introduction, je propose de partager avec le lecteur ma conception du pair-à-pair, non pas comme le sens commun l'entend, mais en tant que domaine de recherche. S'en suivra au chapitre suivant mon positionnement en tant que chercheur à l'intérieur de ce domaine et la structuration du reste du mémoire.

1.1 Histoire subjective du pair-à-pair

Il existe pléthore de travaux essayant de répondre à la question de savoir ce qu'est le pair-à-pair et comment l'aborder. Les taille, technicité et support sont extrêmement variables : livres [72, 70], études en ligne et wikilivres [69, 77], introduction pédagogique [76]...

Face à tout cela, il est difficile de donner une vision des choses qui n'ait pas déjà été proposée ailleurs. C'est pourquoi j'ai choisi de présenter le pair-à-pair en partant de son apparition en tant que phénomène social, vue sous l'angle de mon expérience personnelle. Je ne prétends pas que mon approche soit recommandée pour quelqu'un qui désirerait s'initier au pair-à-pair : il vaut mieux dans ce cas partir d'une des références ci-dessus. Néanmoins, je crois qu'aborder les choses « par le petit bout de la lorgnette », comme je m'apprête à le faire, permet de mieux comprendre le cheminement que j'ai suivi dans mes travaux de recherche, et de faire ainsi émerger ce qui les rassemble.

Ma propre analyse de l'histoire du pair-à-pair est donc la suivante : j'ai tendance à penser que le pair-à-pair est né au moment où c'était la seule solution technique capable de répondre à un besoin social. Plus précisément, remontons à il y a à peu près une dizaine d'années. La démocratisation de l'accès à Internet, couplée à des avancés significatives en traitement du signal (connues du grand public au travers des sigles `MP3` et `DiVX`) commence à rendre possible de récupérer dans des délais raisonnables du « contenu » audio et vidéo.

Problème, d'où récupérer le contenu ? À l'époque (fin des années 90), l'économie numérique n'en est qu'à ses balbutiements. Les médias classiques hésitent encore à franchir le pas numérique, et il n'existe bit [1] de portail permettant l'accès au contenu.

Dans un premier temps, des méthodes relativement anciennes sont employées au sein de cercles restreints d'utilisateurs *underground* : utilisation de *FTPz* (piratage d'un serveur FTP anonyme

1. Néo-grammaticalisation inspirée de l'histoire de la *négation en français*.

afin d'utiliser les ressources en stockage et bande passante du-dit serveur), pièces jointes des forums de discussions (les *alt.binaries* de Usenet), ...

Puis, avec la bulle Internet apparaissent des sites de stockage en ligne. Cette solution est très vite préférée aux *FTPz* : plus besoin de scanner des adresses IP, et risque plus faible de perte des données. L'exemple le plus célèbre de cette époque doit être le site `MySpace.com`, qui n'est pas du tout le site de réseau social actuel, mais offre 300 `Mo` de stockage par compte, sans limite de nombre de comptes et de bande passante [2].

Hélas, un jour, la bulle éclate. Les sites comme `MySpace.com` ferment, ou sont complètement bridés dans le meilleur des cas. L'absence de *business model* autour d'une offre de stockage illimitée et gratuite, et les coûts faramineux engendrés par l'utilisation abusive de la bande passante ne sont certainement pas étrangers à cela. Mais une partie des internautes a pris goût à la facilité d'accès au contenu, et ne veut pas revenir en arrière. Le besoin est là. C'est alors que, au gré des forums, une idée commence à émerger : *Et si on mettait en commun nos ressources de stockage et de bande passante ? Avec suffisamment de participants, on aurait plus de ressources qu'avec n'importe quel serveur, non ?.*

Une première tentative, avec le logiciel *Napster*, a déjà tourné court à cause de problèmes légaux. Malgré tout, poussés par le besoin d'accès à du contenu, des logiciels comme *KaZaA* ou *eDonkey* [27, 28] commencent à voir le jour, tandis que quelque part sur la toile, un informaticien inconnu du nom de Bram Cohen travaille sur un protocole de partage de fichier et utilise des fichiers interdits aux moins de 18 ans pour tester ce qui allait devenir BitTorrent [20] [3]. Nous sommes en 2001 et le pair-à-pair (P2P) naît des cendres des FTPz, de MySpace, et de Napster.

Je dois avouer que lorsque j'ai entendu parler pour la première fois de cette idée d'utiliser les utilisateurs, ma réaction fut de penser *ça ne marchera jamais.* Je ne croyais pas les utilisateurs prêts à offrir leur bande passante, avec toutes les contraintes que cela impliquait. De plus, j'avais à la va-vite estimé les performances d'un tel système à l'aide de la conservation de la bande passante, et j'en avais conclu qu'elles seraient extrêmement mauvaises. Depuis, j'ai eu le temps de réaliser mon erreur et de faire amende honorable, comme j'en parlerai plus en détails au chapitre 3.

Aujourd'hui, le succès des applications de partage de fichiers P2P, qui n'utilisent que les ressources fournies par les utilisateurs, n'est plus à démontrer. L'heure est au développement de nouvelles fonctionnalités, comme par exemple la *lecture en continu*, ou *streaming.* Des applications de diffusion en direct (ou plutôt en léger différé) comme CoolStreaming [78], PPLive [39], SopCast [5], TVants [1], ou UUSee [6] deviennent de plus en plus populaires, tandis que la diffusion de *vidéos-à-la-demande* commence à apparaître [3, 4, 2].

1.2 Définir le pair-à-pair

Maintenant que le contexte est posé, on peut commencer à réfléchir à la question de la définition du pair-à-pair. Pour illustrer le problème, je me permets de raconter l'anecdote suivante : on a beaucoup parlé récemment de la loi dite *DADVSI* relative aux *droits d'auteur et aux droits voisins dans la société de l'information.* L'un des buts de cette loi était de contrer les usages illégaux du P2P, voire les logiciels P2P potentiellement illégaux. Mais pour légiférer, il faut bien définir, et c'est ce qu'essaie de faire l'article 21 qui rend passible de trois ans d'emprisonnement et 300 000 euros d'amende le fait *d'éditer, de mettre à la disposition du public ou de communiquer au public, sciemment et sous quelque forme que ce soit, un logiciel manifestement destiné à la mise à disposition du public non autorisée d'œuvres ou d'objets protégés.*

Cet article a fait couler beaucoup d'encre, notamment parce qu'ainsi rédigé, il incluait la plupart des logiciels de serveurs Web, dont le très connu logiciel Apache. Il a ainsi fallu exclure de manière ad-hoc ces logiciels en rajoutant une clause spéciale. Malgré tout, je trouve intéressante la

2. `http://web.archive.org/web/20001019040949/www.myspace.com/Index.asp`

3. J'étais tombé par hasard sur la page de test de Bram Cohen en 2004, mais je suis incapable de la retrouver aujourd'hui, et je ne peux donc compter que sur ma seule mémoire pour restituer ce fait. Je ne me souviens même plus du titre exact de l'œuvre proposée en téléchargement, mis à part que cela devait être l'opus 4 ou 6 d'une série à thème.

définition du P2P qui ressort de ce texte de loi : un logiciel P2P est destiné à mettre à disposition du contenu, le sens de contenu étant a priori le plus large possible.

Cette première définition, que j'appellerai définition de *motivation*, est bien sûr insuffisante pour définir le P2P, comme la loi DADVSI l'a bien mis en évidence. Il faut donc chercher plus loin. C'est généralement là qu'intervient la définition par *opposition* : contrairement au paradigme classique client/serveur, où chaque constituant du système est soit serveur (distributeur de contenu), soit client (consommateur de contenu), le paradigme P2P considère un système où chacun peut être à la fois client et serveur.

Beaucoup plus ciblée que la définition par motivation, la définition par opposition au modèle client/serveur recèle néanmoins une faille lorsque l'on cherche à préciser ce que signifie *pouvoir être à la fois client et serveur*. Observons par exemple le protocole *ssh*, protocole réseau permettant à un client de se connecter à un serveur. Si j'ouvre une session ssh à partir de mon ordinateur vers l'ordinateur d'un ami, il est clair que je ne fais pas de pair-à-pair. Si cet ami ouvre en retour une session ssh sur mon propre ordinateur, chacun aura potentiellement accès aux ressources de l'autre (chaque ordinateur étant à la fois client et serveur), pourtant cela ne ressemble toujours à du pair-à-pair selon le sens commun. Mais si tout un réseau social se met à utiliser des sessions ssh afin de s'échanger du contenu[4], cela finit par former un réseau pair-à-pair, ou plus précisément *d'ami-à-ami* dans ce cas précis. La question est : où est donc la limite ?

Réciproquement, il est très facile d'imaginer des usages client/serveur pour un logiciel P2P. Ainsi, chaque semaine, j'utilise le client BitTorrent pour récupérer des MP3s d'un ami[5]. Lui possède les fichiers, je les récupère, BitTorrent ne servant qu'à faciliter le transfert : les informations permettant la récupération des fichiers sont concentrées dans un petit fichier `.torrent` qu'il est très facile de produire et de s'échanger par mail. Là encore, la frontière entre paradigme client/serveur et P2P est extrêmement floue.

Il apparaît donc que même si c'est une bonne approximation, la définition par opposition est imparfaite. Tout comme l'on ne peut donc pas définir l'essence du pair-à-pair comme étant liée à une technologie, il n'y a pas de catégorisation architecturale (paradigme) simple où faire rentrer le pair-à-pair.

Tout cela explique pourquoi je me suis forgé au fil du temps ma propre définition du pair-à-pair, qui essaie de capturer au mieux le fait que selon mon expérience, le P2P est avant tout et d'abord une manière décentralisée d'aborder un problème de distribution dans un scénario matériel donné. Autrement dit, si l'on regarde l'évolution du P2P de ses débuts à nos jours, on voit un principe de base rester le même par delà les évolutions technologiques : si l'on veut offrir un service quelconque à un grand nombre d'utilisateurs en n'ayant pas ou peu de serveurs à disposition, la seule solution est de prendre les ressources au seul endroit où elles existent, à savoir chez les utilisateurs (les pairs) eux-mêmes.

Cette constatation amène à une troisième et dernière définition du P2P, que je décrirais comme une définition par *complexité* : un système, un usage, un scénario est de type *pair-à-pair* dès l'instant où la manière de répondre à la question

Qui récupère quoi de la part de qui ?

est non-triviale.

Cette définition peut sembler étrange et contre-intuitive, mais ses avantages sont certains : elle s'affranchit des questions technologiques et transforme une question sémantique (*qu'est-ce que le pair-à-pair ?*) en une problématique plus algorithmique. Ainsi, pour savoir si un assemblage de connections ssh forme un système P2P, la définition que je propose réponds *oui, à condition que le système soit plus complexe qu'une succession de distribution bilatérales indépendantes*[6]. En d'autres termes, on pourra parler de pair-à-pair dès que la recherche d'un fichier donné, ou son transfert, utilise plus d'une connection client/serveur.

On obtient aussi, et c'est cela qui m'intéresse en tant que chercheur, une manière de définir ce

4. C'est le principe de fonctionnement de logiciels existants, comme par exemple *Waste*

5. Je précise qu'il s'agit d'enregistrements de travail des répétitions du groupe de jazz dont je fais partie, donc tout aussi légaux qu'inaudibles pour le public.

6. En exercice, je laisse le lecteur s'amuser à répondre à la question *Skype est-il un logiciel pair-à-pair ?* à l'aide des trois définitions que je viens de donner.

qu'est la recherche en pair-à-pair : essayer de comprendre la question *qui récupère quoi de la part de qui* quand celle-ci devient compliqué, la décomposer en sous-questions plus spécifiques d'un problème donné (recherche et distribution, disponibilité et volatilité, équité et hétérogénéité,...) et essayer d'y répondre de la meilleure manière possible.

1.3 Thématiques de recherche

Maintenant que le sujet de la recherche en pair-à-pair est ouvert, je propose de survoler les différentes thématiques et approches sur lesquels un chercheur en pair-à-pair peut être amener à travailler. Je propose d'employer une classification dichotomique basée sur quatre axes : localisation/distribution, structuré/non-structuré, push/pull, théorique/empirique. Cette classification est loin d'être unique et elle n'est pas parfaite, mais je la trouve relativement simple et pratique.

1.3.1 Localisation et distribution

La première manière de décomposer la question *Qui récupère quoi de la part de qui ?* est de séparer cette question en deux sous-problèmes : localisation et distribution. La phase de localisation consiste à répondre à la question *Qui possède quoi ?* : c'est la phase de localisation. Une solution est par exemple qu'un serveur central surveille tout le contenu présent dans un système, permettant ainsi de répondre à toute recherche. C'est par exemple le principe de fonctionnement des *trackers* BitTorrent [20] ou des serveurs *eDonkey* [28]. Bien sûr, ces solutions sont centralisées, et donc non pair-à-pair (la question *qui possède quoi* est alors triviale). Un premier pas vers la complexité est l'utilisation de plusieurs serveurs au lieu d'un seul (les super-nœuds). On réduit ainsi la concentration d'information, qui peut être dangereuse d'un point de vue de sécurité, mais il faut commencer à réfléchir à la communication entre super-nœuds : on a une solution semi-centralisée, semi-complexe, bref semi-pair-à-pair. Enfin, on peut laisser les pairs se débrouiller totalement entre eux pour savoir qui possède quoi. Parmi ces approches pair-à-pair de la localisation, il y par exemple l'inondation (*flooding*), qui consiste à émettre une demande de contenu et à la faire passer de proche en proche, mais l'approche la plus employée est sans doute celle des *tables de hachage distribuées* [34, 64, 73, 60, 68, 79], qui associent à chaque contenu un pair (voire plusieurs) chargé de savoir qui possède ce contenu.

Une fois le contenu localisé se pose alors le problème de sa distribution effective : comment récupérer le contenu ? C'est un problème qui varie énormément suivant ce que l'on entend par distribution : téléchargement d'un petit ou d'un gros fichier, visualisation d'une chaîne en léger différé, d'une vidéo-à-la-demande... Les subtilités liés aux différents types de distribution seront approfondies dans le chapitre 3. Pour l'instant, on retiendra simplement que le problème exact à résoudre étant variable, la palette des solutions est large : arbres de diffusions, utilisation de files d'attente, échanges réciproques...

La plupart du temps, ces deux problèmes, localisation et distribution, sont indépendants. Par exemple, si la localisation dans un système BitTorrent est historiquement assurée par un tracker centralisé, les versions récentes offrent aussi la possibilité de recourir à une DHT. De même, le choix du protocole de distribution peut s'adapter en fonction du résultat de la localisation : c'est l'approche multi-protocoles, employée par MLDonkey [29].

Bien sûr, il y a des exceptions, et la distinction entre localisation et distribution n'est pas toujours hermétique. Ainsi, pour des raisons de sécurité et de confidentialité, le logiciel Freenet [19] est conçu de telle sorte que la distribution emprunte le même chemin réseau que celui créé au moment de la localisation, rendant les deux problèmes indissociables.

1.3.2 Stucturé, non-structuré

Une autre manière d'aborder la question *qui-quoi-de-qui* est de réfléchir à la manière dont on va résoudre le problème en pratique. De ce point de vue, deux courant majeurs existent : les approches structurées et auto-structurées.

Le principe des approches structurées est de faire en sorte que les pairs se conforment à une structure pré-déterminée. Cette structure va guider les pairs dans leur comportement et leur manière d'échanger des messages ou du contenu. La plupart des tables de hachage distribuées utilisent un algorithme structuré [34, 64, 73, 60, 68, 79]. Des solutions structurées ont également été proposées pour résoudre le problème de la diffusion en léger différé [16, 35].

Les dites structures sont généralement des graphes (arbres, arbres à nœuds intérieurs disjoints [16], graphes de de Brujin [34, 35], graphes réguliers, ...), parfois construits à partir d'une topologie virtuelle (anneau [73, 60], d-tore [64], hypercube [73, 60, 68, 79], ...).

La force des approches structurées est que le comportement du système est déterminé par une structure connue à l'avance, et que celle-ci peut être conçue pour garantir des performances optimales, ou quasi-optimales. Leur faiblesse vient des difficultés à maintenir une structure face à la volatilité des pairs. De plus, beaucoup de structures supposent que les pairs ont des capacités similaires, et à de rares exceptions près [75], elles s'adaptent mal à une population souvent hétérogène dans les faits.

L'autre solution est de laisser (plus ou moins) les pairs se débrouiller, c'est-à-dire les laisser prendre des décisions en fonction de leur propre perception du système. Il n'y a alors plus de structure explicite, mais une pseudo-structure floue et implicite résultant de la somme de choix locaux. C'est pourquoi, d'ailleurs, je préfère le terme *auto-structuré* au terme *non-structuré* qui est généralement employé. Après tout, une structure n'est, par définition, que *la manière dont un ensemble est agencé*. Par exemple, avec toutes les propriétés qu'il possède (degré moyen, composantes connexes, diamètre, ...), le graphe d'Erdös-Rényi, qui est construit à partir de décisions locales (cette arête est-elle dans le graphe ?), est pour moi l'exemple parfait d'auto-structure.

L'approche auto-structurée, dont l'un des exemples les plus représentatifs est certainement BitTorrent, est principalement employée dans la distribution du contenu, que ce soit en partage de fichier ou pour la lecture en continu. Elle est utilisée en localisation dans certains systèmes comme Freenet ou Gnutella, et peut apparaître sous une forme hybridée (comme dit précédemment, aucun cloisonnement n'est complètement rigide) dans certaines DHTs [34, 60].

Par construction, les approches auto-structurées sont adaptées à la volatilité et l'hétérogénéité des pairs. Mais comme leur structure n'est pas connue *a priori*, il est délicat de prédire leurs performances, et la validation doit souvent être faite de manière empirique. Cependant, des avancées ont été faites afin de donner aux systèmes auto-structurés une base théorique [63, 33, 11].

1.3.3 Push et Pull

On peut aussi caractériser, d'un point de vue recherche, la nature d'un système pair-à-pair, en regardant où les décisions sont prises : au niveau de la partie client (*pull*), de la partie serveur (*push*), ou un peu des deux.

Dans une approche pull, le demandeur d'un contenu va lui-même initier la distribution auprès des possesseurs du contenu [7], en décidant éventuellement quelle partie récupérer de qui. C'est le genre d'approche employée dans les principaux systèmes de partage de fichiers actuels (BitTorrent et eDonkey).

À l'inverse, la philosophie push veut que ça soit les possesseurs qui gèrent le bon fonctionnement du système, les demandeurs ayant un rôle plus passif. Les systèmes de distribution basés sur des arbres de diffusion, où le contenu est routé de la source aux destinataires, sont généralement considérés comme typiques du push [16, 35].

Comme on peut le voir dans les exemples précédents, cette classification est généralement associée à la problématique de la distribution. La raison est que les principales solution de localisation, les DHTs (mais aussi les solutions de localisation centralisées), font un peu des deux : les possesseurs transmettent de manière pro-active la description de leur contenu à un tiers, et les demandeurs vont par la suite chercher ces descriptions. Cette approche de type *publish/suscribe* peut malgré tout être interprétée, conceptuellement, comme un cas particulier d'hybridation push/pull.

7. Je rappelle qu'en pair-à-pair, il faut prendre demandeur et possesseur d'un point de vue logique, le possesseur d'une partie de contenu pouvant être simultanément demandeur d'une autre partie de ce même contenu.

Une autre idée reçue (que j'ai volontairement reproduite dans les exemples qui précédent) consiste à associer la distribution push aux systèmes structurés et la distribution pull aux systèmes auto-structurés. Intuitivement, cette association est facile à justifier : en approche pull, chaque demandeur organise lui-même la manière dont il récupère le contenu, et on peut penser qu'une structure trop rigide risquerait d'amoindrir la marge de manœuvre. Réciproquement, laisser un ensemble de possesseurs gérer la distribution d'un contenu semble voué à l'échec sans un minimum de concertation, c'est-à-dire de structure. Et pourtant, les algorithmes de diffusion épidémiques [11], dont je parlerai plus en détails au cours du chapitre 3 sont clairement orientés push (avec une légère rétroaction pull, précisons-le), auto-structurés, et malgré tout promis à un avenir je l'espère brillant.

1.3.4 Théorique, empirique

Au-delà des trois critères que je viens de décrire, qui sont liés aux problématiques propres du pair-à-pair, je crois qu'il est aussi nécessaire de considérer la distinction entre théorie et pratique. C'est une classification (voire un clivage parfois) qui a des origines épistémologiques très lointaines, mais dont les répercussions sur le pair-à-pair sont largement aussi importantes que les distinctions évoquées plus haut. Pour reprendre une définition empruntée à Matthieu Latapy, l'approche théorique (ou plus généralement de la recherche fondamentale) cherche en priorité à *comprendre*, tandis que le but de l'approche empirique est avant tout de *faire* (recherche appliquée).

En étant un peu caricatural, le partisan d'une approche purement empirique dénigrera toute recherche qui n'a pas été validée par une série d'expériences en conditions réelles. C'est par exemple la position défendue par Bram Cohen, le concepteur de BitTorrent [8]. À l'inverse, l'approche purement théorique va chercher à proposer des solutions dont on peut montrer qu'elles doivent marcher, ou des modèles dont on peut penser qu'ils capturent l'essence d'un phénomène. C'est ainsi que procèdent la plupart des solutions structurées citées plus haut, ou encore la modélisation fluide de BitTorrent proposée par Qiu et Srikant [62].

Dans l'absolu, chacune des deux approches prise à l'état pur est incomplète, et c'est seulement combinées qu'elles peuvent exprimer pleinement leur potentiel. Soit en proposant une théorie permettant de comprendre des observations (induction), soit en essayant de falsifier (au sens épistémologique) une théorie à travers une série d'expériences. Ce n'est hélas pas toujours aisé, le cloisonnement croissant des domaines n'aidant pas vraiment à réaliser ce grand écart, et un travail de recherche donné (y compris en pair-à-pair) sera toujours perçu avec une couleur plus empirique ou plus théorique, la nuance exacte dépendant bien sûr de l'œil de l'observateur.

1.3.5 Synthèse

Comme les exemples donnés ci-dessus l'ont illustré, la classification que je propose est loin d'être un cloisonnement rigide, et il faut plutôt la voir comme une boussole permettant d'essayer de s'orienter à l'intérieur du domaine de recherche du pair-à-pair. Les systèmes deviennent de plus en hybrides et mélangent des genres supposés antagonistes. Il n'est plus possible, et ce n'est pas forcément une mauvaise chose, d'isoler complètement les thématiques comme ce fut le cas à une époque où les principaux thèmes abordés étaient les propositions de DHTs (localisation, structuré, publish/suscribe, théorique) et les études de traces réelles (localisation ou distribution suivant la trace, empirique).

8. Cette position est parfaitement apparente si l'on consulte son blog, notamment son billet à charge contre le protocole Avalanche : `http://bramcohen.livejournal.com/20140.html`.

Chapitre 2

Positionnement

S'il apparaît clairement, au travers de mes travaux, que ma principale thématique de recherche est le pair-à-pair, je suis conscient qu'il n'est pas forcément évident d'identifier le fil conducteur qui unifie mon travail, même si les concepts de distribution et d'auto-structure apparaissent souvent dans le cadran de la boussole : dimensionnement générique de la distribution avec la loi de conservation de la bande passante [10] ; distribution épidémique en léger différé, c'est-à-dire auto-structurée et pull [11] ; distribution décentralisée de vidéo-à-la-demande (auto-structurée, avec une phase push et une phase pull) ; modélisation d'auto-structure avec l'ensemble de mes travaux sur les réseaux à préférences [55].

Afin de mieux comprendre mon parcours, voici les trois qualités que j'essaie de réunir autant que possible lorsque je pratique la recherche. Je tiens bien sûr à préciser que ce ne sont en aucun cas les qualités qui définissent un bon chercheur d'une manière générale, car je pense que chaque chercheur a son propre profil, sa propre personnalité, et donc ses propres qualités à développer afin de s'épanouir sur le plan professionnel.

Curiosité J'aime découvrir de nouvelles thématiques. Qui plus est, j'apprécie particulièrement lorsque les sujets que j'attaque n'ont pas encore été étudiés, ou très peu. En effet, cette curiosité me pousse souvent à déborder ou dévier de mes intentions initiales, et j'ai toujours la crainte, quand je découvre des résultats que je trouve nouveaux et intéressants, de découvrir par la suite que ces résultats sont connus depuis longtemps. C'est pourquoi, après un doctorat dont le sujet était le PageRank de Google, sujet d'actualité à l'époque, je me suis tourné quasi-naturellement vers les thématiques émergentes du pair-à-pair, mais en privilégiant la modélisation de la distribution, sujet relativement peu abordé à l'époque compte tenu de son importance.

Écoute Je suis un partisan du discontinuisme scientifique, dans le sens où je ne crois pas que la science évolue selon une progression continue, maîtrisée et quantifiable, mais au contraire qu'elle est la somme de ruptures abruptes. À une plus modeste échelle, qui est la mienne, j'applique cette théorie en m'efforçant de repérer les micro-ruptures qui peuvent passer à ma portée : j'écoute et j'observe, sautant sur l'occasion d'explorer de nouvelles pistes, issues de mon domaine, d'autres domaines, voire de ma sphère privée. L'exemple le plus extrême est celui de la naissance des réseaux à préférences acycliques, relaté dans l'avant-propos, mais la plupart des autres thématiques que j'ai étudiées sont aussi basées sur l'écoute et l'observation des autres. Ainsi, mes déboires personnels de téléchargement en pair-à-pair m'ont amené à étudier les mécanismes de diffusion dans la distribution de fichiers [58]. Quelques années plus tard, écoutant un exposé de Laurent Massoulié, j'ai réalisé que ce que j'avais fait présentait beaucoup de ressemblances avec la diffusion épidémique, ce qui m'a poussé à explorer cette thématique.

Intuition Enfin, je suis incapable de travailler sur un sujet sans en avoir une compréhension intuitive forte, soit initiale, soit acquise au cours des travaux préliminaires. C'est d'ailleurs autant un défaut qu'une qualité. C'est une qualité car l'intuition rend les choses simples, et

facilite la communication de l'essence de ses travaux aux autres. Elle permet de présenter son sujet en escamotant les difficultés techniques, le rendant plus attrayant. C'est un défaut quand les idées qu'elles donnent dépassent la possibilité de les traiter proprement [1]. Faute de temps et de compétence technique adéquate, je traîne ainsi dans mes placards un certain nombre de conjectures dont je *sens* qu'elles sont vraies sans arriver à les résoudre.

Enfin, j'aimerais discuter du positionnement de ma recherche sur le curseur qui va de la théorie (comprendre) à la pratique et à l'empirique (faire). S'il fallait me placer dans l'absolu, ce qui est discutable, j'estime que la recherche que je mène se positionne principalement à un niveau théorique. Personnellement, et à cause de mon parcours à la frontière de l'académie et de l'industrie, je préfère me considérer comme un intermédiaire. C'est une situation agréable lorsqu'elle permet d'apporter un éclairage théorique à un phénomène concret ou de transformer une équation en application concrète. Elle l'est moins lorsque je réalise à quel point mes compétences individuelles, prises séparément, sont faibles comparées aux ténors de chaque *camp*. Mes compétences en programmation ne me permettent guère d'aller au-delà du stade de la simulation, et à ma trousse à outils mathématiques est moins fournie que celle de bien des chercheurs que je connais. Mais j'ai pris le parti de tourner ce que je considère comme une faiblesse personnelle en avantage, et plutôt que de chercher une compétition que je suis sur de perdre dans les deux domaines de la théorie et de la pratique, je préfère me voir comme un passeur entre deux mondes qui idéalement devraient n'en former qu'un.

Organisation du mémoire

Ce mémoire est structuré en trois chapitres. Le chapitre3 s'intéresse aux travaux que j'ai mené autour du problème de la distribution de contenu. C'est un chapitre de synthèse, où j'essaie de donner les idées principales de ces travaux, articulées autour de trois sous-thèmes : la loi de la bande passante (dimensionner), la diffusion épidémique (simplifier), la vidéo-à-la-demande (décentraliser).

Vient ensuite le cœur de ce mémoire, le chapitre 4, qui expose mes travaux sur les réseaux à préférences acycliques. En écrivant ce mémoire, j'avais une volonté très forte de donner une vision globale et unifiée de ce domaine que j'ai découvert il y a maintenant un peu plus de trois ans, c'est pourquoi j'y suis un peu plus exhaustif que pour un simple chapitre de synthèse.

Je termine naturellement par un chapitre conclusif, qui a pour but de replacer mes travaux dans le contexte du P2P à la fois au sens où je l'ai défini, mais aussi dans sa compréhension courante. De cette synthèse ressortent les futurs axes de recherche qui de mon point de vue méritent d'être étudiés afin de pouvoir appréhender pleinement les évolutions à venir autour de la philosophie P2P.

1. « Le succès c'est 1 % d'inspiration et 99 % de transpiration ! », *Albert EINSTEIN*

Chapitre 3

Distribution de contenu

L'objet de ce chapitre est de faire le tour des problématiques liées à la distribution de contenu dans une approche pair-à-pair, en se focalisant sur les thèmes que j'ai personnellement abordé au cours de mes recherches. Je commence par préciser ce que j'entends exactement par *distribution de contenu* (partage de fichier, diffusion en léger différé, vidéo-à-la-demande), en soulignant les enjeux spécifiques et communs de chaque type de distribution. Je parle ensuite de trois problématiques spécifiques : dimensionnement en bande passante (générique à tous les types de distribution), diffusion épidémique (léger différé), et vidéo-à-la-demande décentralisée

3.1 Distribuer

Lorsque je parle de distribution, je pense avant tout à du contenu de grande taille : programmes et OS informatiques, fichiers musicaux, film en DivX, chaîne diffusée en haute définition [1]. Pour ce genre de contenu ayant besoin de grandes ressources réseaux, on peut distinguer grossièrement trois manières d'effectuer la distribution : partage de fichier, diffusion en léger différé et diffusion *à-la-demande*.

3.1.1 Partage de fichier

Historiquement, l'application reine du pair-à-pair en distribution est le partage de fichiers, dont le succès n'est je pense plus à démontrer. Comme dans le cas d'un téléchargement classique (non pair-à-pair), il s'agit d'une application de type *best effort* par nature, et le temps de téléchargement effectif constitue la principale métrique. D'un point de vue réseau, un enjeu est donc d'arriver à utiliser de manière optimale les ressources disponibles, afin de minimiser ce temps de téléchargement. La garantie d'une certaine pérennité du contenu partagé (éviter les téléchargements impossibles) joue également un rôle.

3.1.2 Diffusion *streaming*

Une application de *streaming* a pour but de transmettre un contenu multimedia à des fins de visualisation ou d'écoute. Ce genre d'application n'est pas du tout *best effort*, contrairement au partage de fichier, car des contraintes apparaissent au moment de la lecture du contenu (principalement, il faut s'assurer que la partie du contenu que l'on veut lire soit là quand on on en a besoin). Parmi les métriques communes à toute diffusion, il y a :

La qualité en première a approximation, elle est proportionnelle au débit du flux, même si d'autres paramètres rentrent en compte, comme le choix de l'encodage ou le taux de pertes.

1. La diffusion de contenu de très petite taille (par exemple un système de petites annonces pair-à-pair), du fait de l'inexistence de la bande passante comme paramètre limitant, obéit à des enjeux radicalement différents de ceux que je vais présenter ici.

Le délai d'initialisation c'est le temps qui s'écoule entre le moment où l'on décide de regarder un contenu et celui où celui-ci commence à être lu.

D'un point de vue réseau, l'enjeu est d'arriver à
- maximiser le débit utile, notamment en minimisant les messages de contrôle (*overhead*),
- minimiser les pertes,
- minimiser le délai de transmission (réseau).

Ces trois paramètres sont étroitement reliés entre eux en pair-à-pair. Par exemple, il est généralement possible de réduire le délai de transmission au prix de pertes ou d'un overhead plus élevés. Tout l'art de la conception d'un système de diffusion consiste alors à trouver le bon compromis.

3.1.2.1 léger différé

Le streaming en léger différé (le terme *direct* étant quelque peu présomptueux sur Internet) cherche à diffuser un évènement qui est en train de se produire, par exemple un match de football. En plus des contraintes liées à toute diffusion sur Internet, le principal enjeu du léger différé est d'arriver à minimiser le délai de transmission de bout-en-bout, c'est-à-dire le temps qui s'écoule entre le moment où un but est marqué (et donc où l'on entend ses voisins, qui regardent le match à la télévision, crier) et celui où on peut le voir sur son écran d'ordinateur. D'un point de vue réseau, les enjeux sont les mêmes que pour le streaming en général, mise à part que l'importance du délai de transmission est accrue. La nature même du léger différé crée aussi des avantages et des inconvénients spécifiques. Par exemple, il n'est pas nécessaire de se préoccuper de l'existence du contenu (il est directement injecté dans le système), mais il faut pouvoir satisfaire un grand nombre de personnes regardant un même contenu de manière quasi-synchrone.

3.1.2.2 Vidéo-à-la-demande (VoD)

La vidéo-à-la-demande consiste principalement à associer un système de partage de fichiers à un système de streaming. En plus des contraintes liées à la diffusion, il faut être capable de gérer un catalogue de contenu de grande taille, et en particulier d'assurer sa disponibilité de manière asynchrone, chaque client voulant voir ce qu'il veut quand il veut.

3.2 Dimensionner

La loi de conservation de la bande passante est un outil tellement simple que beaucoup de gens ont tendance à oublier son existence, alors qu'elle permet à peu de frais de comprendre et de dimensionner (approximativement au moins) un système de distribution. Comme je l'ai précisé dans l'introduction, la première fois que je m'y suis intéressé, c'est lorsque j'ai vu les premières discussions sur la possibilité de partager des fichiers en P2P apparaître sur divers forums.

Ma première réaction fut négative : en effet, dans un réseau, si l'on fait abstraction des données de contrôles, des pertes et autres réplications multicast, le transfert de données peut être comparé à un transfert de matière physique : si Alice envoie un litre de musique à Bob, et si tout se passe bien, un litre de musique part de chez Alice, s'écoule dans le réseau, et un litre de musique finit par arriver chez Bob. Le problème, c'est que les tuyaux de sortie et d'entrée n'ont pas la même taille dans la technologie ADSL, qui est celle employée aujourd'hui pour la plupart des accès domestiques, et la capacité de réception est généralement de 4 à 20 fois supérieure à la capacité d'émission. C'est pourquoi j'étais assez crédule quant à l'avenir du pair-à-pair : même si tous les participants ouvraient complètement leurs robinets d'émission, il me semblait que ma baignoire se remplirait toujours moins vite en pair-à-pair que si j'utilisais un serveur capable de saturer mon tuyau d'arrivée.

Cette intuition était fausse. Aujourd'hui, alors que l'on peut télécharger les derniers épisodes des séries en vogue en moins de temps qu'il n'en faut pour faire chauffer une pizza, je me suis forgé une nouvelle intuition, toujours basée uniquement sur la loi de conservation de la bande passante, qui permet de comprendre et de prévoir (dimensionner) le comportement d'un système pair-à-pair.

La clé de mon erreur était que j'avais oublié que dans un réseau pair-à-pair, tout le monde n'est pas simultanément client et serveur (c'est-à-dire *leecher* pour employer le terme consacré).

Il faut tenir compte de ceux qui laissent ouvert leur robinet de sortie sans utiliser celui d'entrée (les *seeders*), et aussi éventuellement, dans le cas de systèmes hybrides, de serveurs dédiés charger d'épauler le système. Au bout du compte, la bonne manière d'écrire la loi de conservation de la bande passante est

$$\sum_{l \in L} d(l) = \min(D_{\max}, U_L + U_S + U_E), \tag{3.1}$$

où L, S et E désignent respectivement la population des leechers, des seeders et des serveurs, U_X l'émission totale de la population X, et $D_{\max}$ la capacité totale de réception des leechers.

Malgré sa simplicité, cette équation mène à des résultats assez fins, dont voici les principaux [10] :

- Pour un système fermé de *streaming* il existe un seuil de scalabilité qui dépend du débit que l'on veut atteindre, de la capacité moyenne d'émission et du rapport entre S et L.
- En dessous de ce seuil de scalabilité, un système hybride permet de démultiplier la capacité initiale des serveurs, l'amplification dépendant des paramètres évoqués ci-dessus.
- pour un système ouvert, avec arrivées et départs, on obtient des résultats similaires en considérant l'intensité du processus d'arrivée et le comportements des pairs à la place de la taille des populations.
- Contrairement à un système fermé, un système ouvert possède une zone de sur-performance, dont la frontière est définie par le comportement des seeders, où les leechers peuvent recevoir à capacité maximale indépendamment de l'intensité d'arrivée. Intuitivement, un système ouvert peut accumuler suffisamment de seeders pour rendre la capacité de transfert arbitrairement grande. Une politique de taux de partage imposé est une bonne manière de se rapprocher de cette zone critique.
- Pour des systèmes à incitation de type tit-for-tat, à la BitTorrent, il existe un seuil de tolérance aux utilisateurs égoïstes (free-riders). Au-delà de ce seuil, le système n'arrive plus à évacuer les free-riders, qui s'accumulent continuellement dans le système.

3.3 Simplifier

Lorsque la diffusion en léger différé est devenu l'objectif à atteindre en pair-à-pair, les courants théoriques et pratiques ont suivi deux chemins radicalement différents : d'un côté, des solutions théoriques ont été proposées à base d'arbres à nœuds intérieurs disjoints [16, 35]. Le problème de ces techniques est que pour diverses raisons que je n'aurais pas la place d'évoquer ici, il n'existe pas à l'heure actuel de client viable basé sur celles-ci. En conséquence, ces approches structurées n'ont pas encore réussi à toucher le grand public.

De l'autre côté, le camp pratique est parti de ce qui marchait déjà, à savoir les applications de partage de fichiers comme BitTorrent, et a cherché à l'adapter à la diffusion. Il en résulte une génération de nouvelles applications comme PPLive [39], qui connaissent un succès croissant auprès du grand public. Mais le problème de cette approche, même si le résultat est fonctionnel, est que BitTorrent n'est pas fait pour diffuser du direct. En particulier, alors qu'un flux direct se doit d'être transmis le plus linéairement possible, l'idéal étant de recevoir les morceaux du flux dans l'ordre dans lequel ils ont été créé, BitTorrent est conçu pour pour mélanger autant que possible l'ordre des morceaux afin de maximiser la disponibilité et d'éviter l'apparition d'un morceau *manquant* [58]. Pour contourner cette contradiction, la solution généralement employée est de regrouper les morceaux successifs en macro-morceaux. Chaque macro-morceau pris individuellement est diffusé *à la BitTorrent*, tandis que la récupération des macro-morceaux successifs se fait de manière linéaire, ce qui permet à la fin de lire le flux dans le bon ordre, macro-morceau après macro-morceau. L'inconvénient est qu'au bout du compte les performances obtenues, notamment en terme de délais d'initialisation et de transmission bout-à-bout, sont bien en-deçà des promesses de la théorie.

Face à cette alternative, une troisième voie, qui a retenu mon attention au cours de mes travaux, est celle de la diffusion épidémique. Si je devais donner une unique raison pour justifier mon choix, cela serait leur simplicité (à ne pas confondre avec trivialité). En effet, alors que les algorithmes

structurés nécessitent la mise en place d'une structure assez complexe et que les applications actuelles rajoutent un niveau supplémentaire à un algorithme existant qui est déjà lui-même assez élaboré, la conception d'une diffusion épidémique est beaucoup plus simple. Lorsqu'un pair a la possibilité technique d'envoyer un morceau du flux à l'un de ses voisins, il doit répondre aux deux questions suivantes

- *À qui doit-il envoyer ?*
- *Quel morceau doit-il choisir d'envoyer ?*

À chaque manière de répondre à ces deux questions (dont on remarquera qu'elles sont très proches du *Qui récupère quoi de la part de qui ?* que j'ai proposé pour définir le pair-à-pair) correspond un algorithme de diffusion épidémique. On peut ainsi générer très facilement un nombre quasi-infini d'algorithmes distincts, et évaluer par une analyse théorique ou des simulations leurs performances [11].

Sans rentrer dans les détails techniques, je dirais à propos de la recherche en cours qu'un consensus semble se dégager quant au choix du morceau (deuxième question) : une fois le destinataire choisi, le plus efficace semble être d'envoyer le dernier morceau utile (*latest useful chunk*), c'est-à-dire le plus récent morceau que la cible ne possède pas encore. Bien que des variations soient encore à l'étude, cette stratégie *latest useful* donne pour l'instant les meilleures performances en terme de délai.

Pour la première question, le choix du destinataire, la question est encore loin d'être tranchée : on sait que dans certains cas, choisir le destinataire simplement au hasard donne des performances quasi-optimales sur un plan théorique, mais les simulations suggèrent qu'il y a beaucoup de marge dans ce *quasi*. Les recherches actuelles (menées à Orange Labs, mais aussi par d'autres équipes en France et en Europe) visent donc à faire mieux que le choix aléatoire, sachant que pour choisir autrement qu'au hasard, il faut donner à l'émetteur plus d'informations, et qu'il y a un compromis à trouver entre la quantité d'information à faire parvenir à l'émetteur au moment de son choix et la qualité du choix qui en résulte. À l'extrême, on pourrait imaginer un mécanisme où les émetteurs communiquent de manière à se synchroniser et parviennent à créer une diffusion optimale, mais le coût d'une telle technique en terme de messages de contrôle (*overhead*) risquerait fort d'être prohibitif.

3.4 Décentraliser

Enfin, j'aimerais clore ce chapitre sur la distribution de contenu en évoquant brièvement le problème de la vidéo-à-la-demande décentralisée, que j'ai eu l'occasion d'aborder sous l'impulsion de Laurent Viennot [12, 13]. Ce problème, originalement proposé par Suh *et al.* [74], utilise le fait que de plus en plus de foyers sont équipés d'adjoints de poste de télévision (*set-top-boxes*), qui ont entre autres caractéristiques un disque dur et un accès Internet. D'où cette idée très simple : puisqu'un service de vidéo-à-la-demande nécessite essentiellement pour fonctionner de la capacité de stockage et de la bande passante, pourquoi ne pas construire un service qui repose sur les ressources des *set-top-boxes* ?

L'intérêt de ce sujet est qu'au-delà des problèmes classiques que l'on rencontre en distribution (bande passante et délai d'initialisation par exemple), on voit apparaître l'existence d'un compromis entre bande passante, capacité de stockage et taux d'échec. Ainsi, si l'on stocke les fichiers vidéos avec une redondance faible, plus la bande passante disponible est proche de la limite nécessaire au bon fonctionnement du service (loi de conservation), plus le risque est grand qu'en pratique, au moment où il faut distribuer une vidéo, les set-top-boxes possédant cette vidéo aient déjà épuisé les ressources réseau nécessaires, entraînant l'échec de la distribution. Pour diminuer le risque d'échec, il faut soit augmenter les capacités réseau, soit augmenter la redondance, et donc mécaniquement diminuer la capacité de stockage disponible [2].

Une autre spécificité de ce problème est que bien que des solutions pratiques très simples, à base de choix aléatoires, puissent être mise en œuvre pour le résoudre de manière presque optimale, la théorie d'allocation sous-jacente est très subtile et assurément non-triviale.

2. La manière dont les vidéos sont découpées intervient aussi [12, 13].

Chapitre 4

Réseaux à préférences acycliques

L'objectif de ce chapitre est de présenter une contribution à laquelle je tiens tout particulièrement, à savoir le modèle des *réseaux à préférences*. J'ai proposé ce modèle afin de mieux comprendre la dynamique des collaborations entre des pairs qui agissent indépendamment selon leurs propres préférences.

Ce chapitre est structuré en quatre partie : je propose tout d'abord de faire donner un bref aperçu de la théorie des mariages stables, d'où est tirée le modèle des réseaux à préférences. J'introduit ensuite les grandes bases : le formalisme employé, le grand théorème des préférences acycliques, qui est au centre de la théorie, et enfin une zoologie de ces fameuses préférences acycliques. La troisième partie est ensuite consacré à une étude fine des propriétés d'auto-stabilisation révélées par le grand théorème, tandis que la dernière partie est consacrée aux méthodes permettant de décrire la configuration stable d'un réseau à préférences acycliques.

4.1 Origine

L'origine des réseaux à préférences est la théorie des *mariages stables* proposée par Gale et Shapley. La plus ludique version de ce problème apparaît dès l'article original de 1962 [36] : il s'agit de marier des hommes et des femmes en faisant en sorte d'éviter tout risque d'adultère. Formellement, une instance du problème des mariages stables est constituée de deux ensembles H et F (les hommes et les femmes), chacun classant strictement un sous-ensemble des membres du sexe opposé. Une configuration de cette instance est tout simplement un couplage entre H et F. Elle est instable s'il existe un couple (h, f) n'appartenant pas à la configuration, tel que h préfère f à sa partenaire dans la configuration (ou si h est célibataire, que h accepte d'être avec f), et réciproquement. Si cela se produit, l'adultère n'est pas loin, car h et f ont tout deux intérêt à se mettre ensemble, même si cela signifie la fin de leur couple actuel. À partir de cette définition, l'objectif de la théorie est d'identifier les configurations stables d'une instance donnée. Il a été prouvé dès 1962 que si les préférences sont strictes (pas d'hésitations dans les classements), il existe toujours une configuration stable, que l'on peut facilement obtenir par l'*algorithme du bal* [36].

Une première extension du problème est celui du *recrutement des Maîtres de Conférences* (*college admission* ou *hospitals/residents* en anglais [66, 41]). Dans cette variante, les membres de l'un des ensembles peuvent choisir plusieurs représentants de l'ensemble opposé. Par exemple, une université peut ouvrir plusieurs postes de Maître de Conférence chaque année, mais un futur maître de conférence n'ira qu'à une seule université. Comme pour le problème du mariage stable, il est toujours possible d'obtenir une configuration stable, et certains algorithmes proposés par Gale et Shapley sont toujours en vigueur dans nombre de processus de recrutement.

Mais pour le P2P, l'extension intéressante est celle où il n'y a qu'une seule catégorie de participants. Cette variante unisexe est connue sous le nom du problème des *camarades de chambre* (roommates) dans le cas du couplage simple, et b-couplage (b-matching) dans le cas général, b

étant la notation standard pour le vecteur décrivant les quotas du multi-couplage [30, 40, 42]. Le mariage unisexe trouve des applications dans de nombreux domaines, l'une des variantes les plus actives étant les échanges de donneurs de rein (pairwise kidneys exchange program) [67, 80]. La principale difficulté par rapport aux mariages sexués est qu'il n'est plus possible de garantir l'existence d'une configuration stable, ce qui peut s'avérer problématique, en particulier quand il s'agit de dons d'organe.

Réinventer la roue ?

Intuitivement, l'intérêt des théories de mariage pour le P2P est assez évident : si l'on a un ensemble de pairs, chacun possédant une liste de voisins avec lesquels il est susceptible d'interagir, la théorie devrait pouvoir nous éclairer sur ce qui se passe en supposant que chaque pair agit en fonction de son propre intérêt (i.e. de ses propres préférences).

Comme les théories des mariages existent depuis près de 50 ans, on pourrait penser qu'il suffit de prendre la littérature existante pour avoir l'outil idéal de modélisation P2P. En pratique, les choses ne sont pas aussi simples : les théories des mariages offrent effectivement un bagage théorique extrêmement puissant, mais les spécificités liées au P2P nécessitent de voir le problème sous un nouvel angle, celui des réseaux à préférences, qui sont l'objet de ce chapitre.

Plus précisément, les principaux changements apportés par les réseaux à préférences sont les suivants :

- les préférences des problèmes classiques de b-couplage sont souvent arbitraires, alors qu'en P2P, la plupart des préférences sont dérivées à partir de mesures objectives ;
- l'existence de configurations stables et les moyens de les obtenir sont une composante essentielle des problèmes de b-couplage ; à l'inverse, la plupart des préférences P2P bénéficient d'un puissant théorème d'existence/unicité/convergence qui rend ces questions hors de propos ;
- les questions de dynamique et de performance des configurations sont un aspect fondamental pour le P2P, et doivent donc être mises en avant.

4.2 Les bases

L'objet de cette section est de présenter le domaine des réseaux à préférences, dans ses grandes lignes, damé et balisé après trois années de tâtonnements : les notations et définitions, telles qu'elles se sont stabilisées au fil du temps (§ 4.2.1) ; le grand théorème des préférences acycliques, qui malgré sa simplicité est la colonne vertébrale du modèle (§ 4.2.2) ; et enfin, la zoologie des préférences acycliques, qui permet de voir comment relier le modèle à des situations réelles (§ 4.2.3).

4.2.1 Notations et définitions

Comme indiqué dans l'introduction, les réseaux à préférences sont issus de la théorie des b-couplages [17, 22], et les notations s'en ressentent, même si un sous-vocabulaire spécifique a été introduit au fil des travaux [49, 33, 32, 53, 54]. En ce qui concerne la version française en particulier, le vocabulaire souvent imagé doit beaucoup à l'article de vulgarisation *Mariages Stables* de Dominique Dumont [26].

Ainsi, une instance de réseau à préférences est constituée d'un ensemble de pairs, d'un graphe d'acceptabilité, de préférences et de quotas. Une configuration est un ensemble d'interactions, dont la dynamique est décrite par des initiatives de pairs.

4.2.1.1 Réseau à préférences

Le réseau à préférences lui-même est donc constitué d'un ensemble P de n pairs (ou nœuds) et d'un graphe $G = (P, E)$. Les préférences sont indiquées par une matrice de valeurs m et les quotas par un vecteur b.

Le graphe d'acceptabilité $G = (P, E)$ est un graphe non-orienté et non réflexif. Il décrit les compatibilités : deux pairs i et j peuvent interagir si, et seulement si, (ssi) $\{i, j\} \in E$. On dit alors

que i est *acceptable* pour j, et réciproquement. Par exemple, G peut représenter la connaissance des autres pairs (il est rare qu'un pair connaisse tous les autres pairs d'un réseau, sa vision est généralement limitée à un certain nombre de *voisins*), ou encore l'existence d'un intérêt commun (la recherche d'un même fichier, l'appartenance à un même groupe, ...), ou tout simplement une structure *overlay* imposée. Le graphe G peut a priori être quelconque, cependant nous userons souvent dans ce chapitre des graphes Erdös-Rényi $\mathcal{G}(n,p)$, où chaque arête existe avec probabilité p indépendamment des autres (le degré moyen est ainsi $d = p(n-1)$). Ce choix est justifié par les facilités qu'offrent les graphes $\mathcal{G}(n,p)$ quant à l'analyse théorique des propriétés des réseaux à préférences.

La matrice de valeurs m indique l'intérêt que les pairs ont les uns envers les autres. Elle détermine ainsi les préférences. Pour toute paire acceptable de pairs $\{i,j\} \in E$, $m(i,j)$ est la valeur que i donne à j. Sauf indication contraire, on considère que les basses valeurs sont préférées [1]. Ainsi, $m(i,j) < m(i,k)$ signifie que i préfère j à k. Une hypothèse que l'on fait tout au long de ce chapitre est qu'il n'y a jamais d'égalité, c'est-à-dire que pour chaque ligne de m, les valeurs acceptables sont deux à deux distinctes [2]. La matrice m peut posséder certaines propriétés particulières qui seront décrites au § 4.2.3. Je précise également que s'il est toujours possible, et pratique, de supposer l'existence d'une matrice de valeurs m, cela n'est pas indispensable d'un point de vue théorique, car seuls les ordres (les préférences) induits par m importent [32, 7]. Il n'est pas nécessaire que les pairs connaissent m. Dans le pire des cas, il est même possible qu'un pair ne connaisse pas les valeurs de sa propre ligne. Cependant, afin de garder un sens au concept de matrice de valeurs, on suppose qu'il est toujours possible pour un pair de comparer les valeurs de deux voisins en les contactant, par exemple au cours d'une *initiative* (voir plus loin).

Enfin, le vecteur des quotas, b, limite le nombre de collaborations : un pair i ne peut collaborer avec plus de $b(i)$ voisins simultanément. On peut bien sûr toujours supposer que $b(i)$ n'est pas plus grand que le degré de i dans G. Tout comme G est souvent supposé Erdös-Rényi, b est généralement supposé constant lorsque l'on veut analyser un réseau, même si toute distribution est autorisée a priori.

4.2.1.2 Configurations

La configuration d'un système décrit l'état des collaborations. Formellement, une configuration C est un sous-ensemble de E. Les voisins $C(i)$ de i dans C sont les *partenaires* actuels de i. Si $c(i)$ désigne le nombre de partenaires de i dans C, les quotas impliquent $c(i) \leqslant b(i)$. Si $c(i) < b(i)$, on dit que i est *sous-couplé*.

Une configuration peut évoluer par la résolution d'arêtes *adultères* (le terme est un hommage à la théorie originale des mariages). Une arête acceptable e est adultère pour une configuration C si elle n'appartient pas à C et si chaque pair de l'arête a intérêt à établir la collaboration e, même si cela implique d'abandonner une collaboration de C pour satisfaire les quotas. Formellement, une arête $e = \{i,j\} \in E \setminus C$ est adultère ssi :

- $c(i) < b(i)$ (i est sous-couplé), ou bien $\exists k \in C(i), m(i,j) < m(i,k)$ (i préfère j à l'un de ses partenaires) ;
- $c(j) < b(j)$ ou bien $\exists k \in C(j), m(j,i) < m(j,k)$ (condition symétrique).

Un pair qui est adjacent à au moins une arête adultère est *éligible* (il est susceptible de modifier la configuration). Une configuration qui ne possède pas d'arête adultère (ou de manière équivalente pas de pair éligible) est *stable*.

1. Ce choix est purement conventionnel et n'affecte aucunement les résultats présentés ici. Il est naturel lorsque l'on considère des latences ou des distances. Dans le cas de valeurs comme la bande passante, ou plus généralement dans le cas de capacités, cela serait plutôt l'inverse.

2. Les cas d'égalités dans les préférences sont une source de problèmes que je ne tiens pas à évoquer dans ce chapitre [41, 42, 43, 52, 51, 65]. On supposera donc qu'une égalité peut toujours être brisée, par exemple en attribuant à chaque pair un identifiant unique afin de départager toute ambiguïté.

4.2.1.3 Initiatives

Les arêtes adultères sont à la base de la dynamique des configurations. Plus précisément, l'évolution vient toujours d'un pair éligible qui essaie de résoudre une arête adultère à laquelle il appartient. Ce processus, supposé atomique, est appelé *initiative*. Une initiative est *active* si elle se traduit par la résolution d'une arête adultère.

Formellement, l'initiative est une « fonction » de sélection (qui peut dépendre du temps, de l'état du système, être aléatoire, ...) qui associe à chaque pair i de V un voisin à « essayer », ou plus formellement une arête de $E \setminus C$ incidente à i. Si par défaut, l'ensemble de ces arêtes peuvent être considérées par cette fonction de sélection, dans le cas où les pairs ont la connaissance des arêtes adultères auxquelles ils appartiennent, il est possible de restreindre la sélection à ces seules arêtes.

Des exemples d'initiatives sont la sélection *meilleur partenaire* (parmi les arêtes adultères), *décrémentale* (choix en *round-robin* parmi la liste des voisins), ou simplement aléatoire. Remarquons que les différents types de sélection nécessitent, de manière implicite, plus ou moins de connaissances : ainsi, *meilleur partenaire* et *décrémentale* nécessitent d'être capable de trier ses voisins (et même de repérer les arêtes adultères pour la première), alors qu'une sélection purement *aléatoire* nécessite juste d'être capable d'évaluer un voisin à la volée. Enfin, il est toujours possible d'hybrider plusieurs sélections afin d'obtenir une sélection plus performante. Un exemple de sélection *hybride* sera décrit au § 4.3.2 (d'après [54]).

Partant d'une configuration initiale donnée, l'évolution d'un système est donc décrite par la suite des initiatives réalisées par les pairs. Une suite classique est la suite *Round-Robin*, adaptée pour modéliser un comportement mono-périodique des pairs, ou la suite *uniformément aléatoire*, utilisée pour modéliser un processus Poissonnien homogène (chaque pair suit un processus de Poisson i.i.d.). Enfin, il y a les suites *adversariales*, qui essaient de se comporter de la « pire » façon possible.

Par commodité, le temps est directement mesuré à l'aune de la suite des initiatives (il n'est ainsi pas nécessaire d'introduire explicitement des processus temporels d'initiatives). Il est malgré tout possible de considérer plusieurs mesures suivant le contexte que l'on considère. Par exemple, si l'on se place dans le contexte de l'auto-stabilisation au sens de Dijkstra [23], on comptera seulement les initiatives actives, et les suites seront découpées en *rounds* (un round est une sous-suite telle que chaque pair éligible au début du round prend une initiative ou devient non-éligible au cours de ce round). Pour les simulations, je préfère définir l'unité de temps (u.t.) par n initiatives (actives ou non). Ainsi, chaque initiative atomique prend $\frac{1}{n}$ u.t., et au bout de t u.t., le nombre d'initiatives par pair est en moyenne de t. Lorsque la suite d'initiatives est round-robin, round et unité de temps peuvent être considérés comme synonymes, mais ce n'est pas vrai dans le cas général.

4.2.1.4 Stabilité locale

La notion d'initiative entraîne une définition alternative de la stabilité : une configuration C est stable si la seule configuration atteignable par initiatives à partir de C est C elle-même. Il est facile de vérifier que pour des initiatives arbitraires, cette définition est équivalente à celle donnée en § 4.2.1.2 (pas d'arête adultère). L'intérêt d'une approche au sens des initiatives de la stabilité est qu'elle permet une définition locale : une arête e d'une configuration C est *stable* ssi e existe dans toutes les configurations atteignables à partir de C. Écrit autrement, une arête stable est une collaboration que les initiatives ne peuvent rompre.

Lorsqu'un pair est incident à une arête stable, son degré de liberté s'en trouve diminué. Le quota libre d'un pair i, noté $b'(i)$, est le quota $b(i)$ auquel l'on retranche le nombre d'arêtes stables incidentes à i. Les quotas libres forment un vecteur qui est fonction de la configuration et qui ne fait que décroître avec le temps. Il est ainsi possible d'étendre la définition de stabilité au niveau des pairs : un pair i est *stable* (ou *inactivé*) si $b'(i) = 0$ ou si i partage une arête stable avec tous ses voisins non-stables.

Enfin, une dernière notion fondamentale à propos de la stabilité est celle de *chaleur*. Dans une configuration donnée C, une arête $e = \{i, j\}$ acceptable mais hors de C ($e \in E \setminus C$) est chaude

ssi :
- j appartient aux $b'(i)$ meilleurs voisins non-stables de i,
- i appartient aux $b'(j)$ meilleurs voisins non-stables de j.

Il existe de même une définition plus stricte de la chaleur : une arête acceptable $e = \{i, j\}$ entre deux pairs non-stables qui ne collaborent pas est dite bouillante pour C ssi i est le meilleur voisin non-stable de j et vice versa.

Par extension, un pair sera dit chaud (resp. bouillant) s'il est adjacent à une arête chaude (resp. bouillante).

Intuitivement, les arêtes chaudes, et plus encore bouillantes, sont des arêtes super-adultères, qui n'attendent qu'une initiative bien placée pour devenir des arêtes stables. Elles jouent un rôle central dans les réseaux à préférences acycliques, à cause du lemme 1, qui est utilisé dans beaucoup de démonstrations :

Lemme 1 ([49, 32]) *Si C est une configuration non-stable d'un réseau à préférences acycliques, alors il existe au moins une arête bouillante par rapport à C.*

Bien évidemment, ce lemme n'a de sens qu'une fois définie la notion d'acyclicité. C'est d'ailleurs l'objet du reste de cette section sur les bases des réseaux à préférences.

4.2.2 Grand théorème des préférences acycliques

Un *cycle de préférences*, ou *cycle de Kieschnick*, est un cycle de $k \geqslant 3$ pairs $i_1, \ldots, i_k$ tel que chaque pair préfère son successeur à son prédécesseur : i_1 préfère i_2 à i_k, i_2 préfère i_3 à i_1, ..., i_k préfère i_1 à i_{k-1} (ou encore, exprimé en terme de valeurs, $m(i_1, i_2) < m(i_1, i_k)$, $m(i_2, i_3) < m(i_2, i_1)$, ..., $m(i_k, i_1) < m(i_k, i_{k-1})$).

L'objet principal de ce chapitre est les réseaux à préférences acycliques, c'est-à-dire qui ne contiennent pas de cycle de Kieschnick. Remarquons que l'acyclicité est entièrement définie par m, et qu'elle ne dépend ni des quotas, ni des initiatives. De même, si m produit des préférences acycliques pour une acceptabilité complète, alors le réseau sera acyclique pour tout graphe d'acceptabilité G. Par convention, une telle matrice de valeurs m est également appelée acyclique. Une classification des matrices acycliques, et le lien avec les systèmes P2P, sont proposés en § 4.2.3.

Les réseaux à préférences acycliques sont caractérisés par une propriété qui les rend uniques au sein des réseaux à préférences :

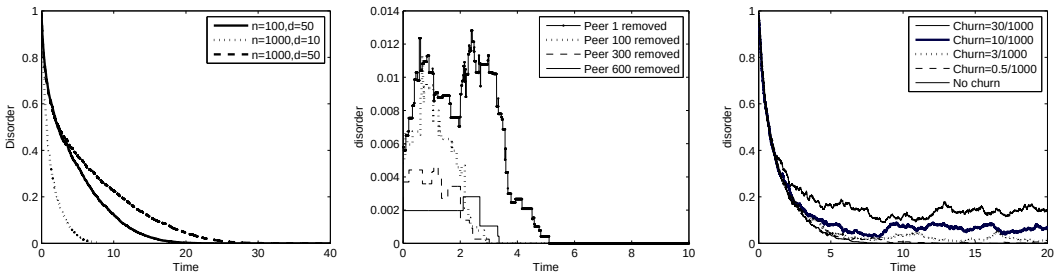
Théorème 1 ([32]) *Un réseau à préférences acycliques admet une, et une seule, configuration stable. De plus, il est auto-stabilisant par initiatives.*

Idée de la preuve : La preuve complète est disponible dans [32], mais j'en donne ici l'idée, à cause de l'importance de ce théorème et du caractère typique des techniques employées. L'on va ici procéder en deux étapes : d'abord montrer l'auto-stabilisation (ce qui prouvera l'existence d'une configuration stable), puis l'unicité.

L'auto-stabilisation vient de la propriété suivante : quelque soit la suite d'initiatives considérée et la configuration de départ, la suite des configurations correspondantes est irréversible. C'est-à-dire que si le système se trouvait dans une configuration C_1 dans le passé et se trouve dans une configuration $C_2 \neq C_1$, alors C_1 n'est pas atteignable à partir de C_2. L'irréversibilité se démontre par l'absurde, en extrayant un cycle de Kieschnick à partir d'un cycle de configurations. Comme n est supposé fini, le nombre de configurations possibles l'est également, ce qui veut dire que toute trajectoire de configurations débouche nécessairement sur une configuration qui n'évolue plus, i.e. une configuration stable.

L'unicité se prouve également par contradiction : si l'on considère deux configurations stables distinctes A et B d'un même système, et un pair i dont les collaborations diffèrent de A à B, alors on peut construire un cycle de Kieschnick en partant de i (même si i n'appartient pas nécessairement au cycle que l'on obtient au final).

□



(a) Cas statique (b) Altération atomique (c) Agitation continue

FIGURE 4.1 – L'auto-stabilisation en pratique (d'après [33]).

J'insiste encore une fois sur le caractère fondamental de ce théorème d'existence/unicité/auto-stabilisation, car il permet de s'affranchir de l'étude de stabilité faite pour les autres types de préférences, et de se concentrer plus finement sur les configurations proprement dites.

La première application de ce théorème est apparue dans l'article *Stratification*[3] *in P2P Networks : Application to BitTorrent* [33]. Dans cet article, qui cherchait à modéliser un réseau BitTorrent comme un réseau à préférences acycliques (j'en parlerai plus en détail dans la suite du chapitre), nous avons mesuré la convergence effective entre configuration réelle et stable. Afin d'estimer l'importance pratique de l'auto-stabilisation, nous avions considéré trois types de scénarios (cf Figure 4.1) :

Statique (4.1a) Partant de la configuration vide (souvent notée $C_\emptyset$), nous avons observé la convergence vers la configuration stable. Nous avons pu observer une bonne convergence dans tous les cas, tout en remarquant que les paramètres du système, et le degré moyen du graphe d'acceptabilité en particulier, jouaient un rôle majeur ;

Altération atomique (4.1b) Un deuxième scénario élémentaire consiste à enlever un pair à partir d'une configuration stable (ce qui modifie la configuration stable), et refaire converger le système. On observe que la configuration réelle ne s'éloigne guère de la configuration stable, et finit par re-converger, même si à cause d'un éventuel effet domino, il se peut que le temps de convergence soit comparable au cas statique.

Agitation continue (*Continuous churn*) (4.1c) Enfin, un troisième scénario consiste à faire partir et arriver des pairs dans le système selon une certaine intensité. Ce scénario permet de constater que la configuration réelle n'arrive pas toujours à rattraper l'évolution de la configuration stable, surtout quand l'agitation est extrême. Cependant, l'auto-stabilisation fait en sorte que la configuration réelle reste malgré tout assez proche de la configuration stable, la *distance* entre les deux étant grossièrement proportionnelle à l'agitation.

À la suite de cette étude, j'ai commencé à travailler en partant du principe qu'à la condition que la convergence du système soit suffisamment grande devant l'évolution de celui-ci, la configuration stable devait être une bonne approximation des configurations réelles. C'est l'image du *ressort* (cf Figure 4.2). Cette image nous suggère de décomposer l'étude des systèmes à préférences acycliques en deux problèmes distincts :

Quelle est la vitesse de convergence ? En effet, le théorème 1 ne nous donne que le nombre de configurations possibles comme borne. Ce nombre croît de manière factorielle [18] et offre donc un intérêt pratique assez limité. Une première zone d'étude consiste donc à caractériser plus finement la *raideur* effective du ressort.

Quelles sont les propriétés des configurations stables ? La question précédente doit permettre de déterminer dans quelle mesure une configuration stable est une bonne approxi-

3. Beaucoup de personnes me demandent pourquoi *stratification*, au lieu de *clusterisation* par exemple. Et bien tout d'abord, parce que *clusterisation* est déjà sémantiquement surchargé en informatique ; ensuite parce que l'image des strates est au moins aussi parlante ; enfin parce que le mot est joli.

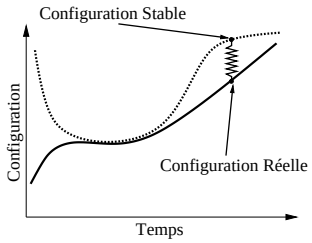


FIGURE 4.2 – L'image du ressort : l'auto-stabilisation peut être vu comme une attirance continue de la configuration actuelle vers la configuration stable, comme si un ressort reliait ces deux configurations. Ce principe vaut même si la configuration stable bouge dans le temps.

mation d'une configuration réelle. Quand c'est le cas, les propriétés génériques des solutions stables peuvent permettre d'estimer les performances du système.

Ces deux questions vont faire l'objet des deux prochaines sections de ce chapitre. Mais avant d'y répondre, je propose de regarder d'un peu plus près ces préférences acycliques dont je n'ai jusqu'à présent donné qu'une définition sommaire et peu pratique.

4.2.3 Zoologie des préférences acycliques

Parmi toutes les matrices réelles possibles, peu sont acycliques. En fait, si l'on prend une matrice dont les coefficients sont choisis au hasard, il est très probable que les préférences qui en résultent contiennent au moins un cycle de longueur 3 [4]. D'un autre côté, beaucoup de propriétés matricielles sont synonymes d'acyclicité. Par exemple ([32]) :

Les matrices globales Une matrice de valeurs m est globale si toutes ses lignes sont identiques si l'on se restreint aux entrées acceptables. Intuitivement, c'est un cas particulier de matrice de rang 1, et formellement, on a $m(i,k) = m(j,k)$ pour tout les i, j qui acceptent k. Les préférences correspondantes, également appelées globales, reflètent un ordre total sur les pairs. On peut d'ailleurs remarquer qu'à cause de cet ordre total, on peut dire qu'à acceptabilité et permutation des pairs près, il n'y a qu'une seule préférence globale.

Les matrices symétriques Si m est symétrique sur ses entrées acceptables ($m(i,j) = m(j,i)$ pour toute paire acceptable) m est acyclique. Une propriété intéressante des matrices symétriques est qu'elles décrivent l'ensemble des préférences acycliques (cf [32, 7]). Le lien entre matrices acycliques, symétriques, globales, et les préférences correspondantes est illustré Figure 4.3 [5].

Combinaison linéaire de matrices globales et symétriques Les matrices obtenues sont acycliques tant qu'elles ne génèrent pas d'égalité (mais comme indiqué en § 4.2.1.1, il est toujours possible de s'affranchir des cas d'égalité). En revanche, une combinaison linéaire de matrices acycliques n'est pas forcément acyclique : si les matrices d'entrée ne sont pas déjà elles-mêmes combinaison linéaire de matrices symétriques et globales, il faut les symétriser, ce qui se fait assez facilement car la preuve de surjection des matrices symétriques dans les préférences acycliques est constructive (cf [32, 7]).

4.2.3.1 Préférences adaptées au P2P

Après avoir esquissé ce qui fait que des préférences sont acycliques, je propose maintenant de nous recentrer dans le contexte du pair-à-pair, et de considérer les préférences pair-à-pair acycliques.

4. La probabilité que trois pairs mutuellement acceptables forment un cycle est de $\frac{1}{4}$, pour des valeurs aléatoires uniformes. La probabilité de ne pas avoir de tel cycle est donc au plus de $(\frac{3}{4})^{\lfloor \frac{n}{3} \rfloor}$ ($(\frac{3}{4})^{\binom{n}{3}}$ si l'on néglige de possibles corrélations), ce qui ne vaut pas grand chose dès que n est assez grand.

5. La Figure 4.3 est bien sûr incomplète. Il manque par exemple les combinaisons linéaires en général, et les matrices complémentaires en particulier, ainsi que la génération de l'ensemble des préférences acycliques par des permutations des entrées d'une matrice symétrique à coefficients distincts. Mais la lisibilité graphique s'en trouverait affectée.

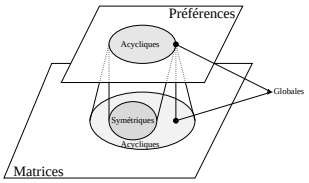


FIGURE 4.3 – Correspondances entre matrices et préférences : les matrices symétriques et globales forment deux sous-ensembles distincts de matrices acycliques, mais les matrices symétriques peuvent décrire l'ensemble des préférences acycliques (ce qui inclut les préférences globales).

Capacités Un pair dans un réseau P2P possède de nombreuses caractéristiques scalaires intrinsèques qui peuvent créer des préférences : bande passante d'accès, capacité de stockage ou de calcul, *uptime* moyen,... Si l'on prend l'exemple du protocole de partage de fichiers BitTorrent [20], un algorithme *donnant-donnant* fait qu'un pair tend à collaborer de préférence avec ses voisins qui lui fournissent la plus grande vitesse de téléchargement. Cela peut être vu en première approximation comme un réseau à préférences où la valeur est la bande passante d'émission divisée par le quota (le nombre d'émissions simultanées).

Proximités Toutes les valeurs qui peuvent être assimilées à une sorte de distance (physique ou virtuelle) ou de similarité sont symétriques par nature. Ainsi, beaucoup de systèmes P2P essaient de minimiser les latences [6], comme par exemple la DHT Pastry [68] ou les applications de jeu en ligne en temps réel [50]. De même, les jeux massivement multi-joueurs en ligne (MMOG) doivent connecter ensemble les joueurs proches dans un certain espace virtuel [46, 45, 21]. Certains auteurs proposent également de connecter les participants d'un système de partage de fichiers en fonction de leurs intérêts communs [71, 48], ce qui reste une mesure symétrique. Le *co-uptime*, c'est-à-dire le temps (moyen) d'activité commune, est un dernier exemple de mesure symétrique ayant un intérêt pour des applications collaboratives.

Complémentarités La mesure des différences entre les ressources des participants peut également s'avérer intéressante. Par exemple, dans une application de stockage distribué de fichiers, il est utile de trouver des machines qui sont allumées quand ma propre machine est éteinte, afin de garder mes données disponibles à tout moment. La mesure correspondante est l'uptime complémentaire. De même, dans un système comme BitTorrent, tous les participants cherchent à avoir le même fichier, qui est divisé en blocs [7]. Trouver les voisins qui ont le plus de blocs que je n'ai pas encore obtenus a un intérêt. Toutes ces préférences dites complémentaires sont un cas particulier de combinaison linéaire d'une matrice globale et d'une matrice symétrique, et sont donc acycliques [32].

4.2.3.2 Classes de préférences étudiées

Comme nous venons de le voir, il existe de nombreuses préférences acycliques présentant un intérêt du point de vue du P2P. Au cours de mes travaux, je me suis concentré sur quatre classes de préférences acycliques, ce qui évite de se disperser dans un labyrinthe de cas particuliers tout en donnant une idée précise des liens entre préférences et propriétés des systèmes.

Globales Bien que représentant un sous-ensemble négligeable (en terme de cardinalité) des préférences acycliques, les préférences globales sont parmi les plus importantes des préférences acycliques (elles modélisent les capacités), et doivent recevoir une attention particulière. Comme il n'existe en fait qu'un unique système de préférences globales, induit par un ordre total sur les pairs, j'utiliserai cet ordre au lieu d'expliciter la matrice des valeurs m. Les pairs seront ainsi étiquetés de 1 à n, 1 étant le *meilleur* (il est préféré par tous ses voisins acceptables), et ainsi de suite...

6. Se pose bien sûr la question de savoir si les latences sont des distances, ou même symétriques. Quoiqu'il en soit, empiriquement, elles ne produisent pas de cycle de Kieschnick.

7. Il paraît même que parfois, certains blocs pourraient être plus rares que d'autres [58].

Géométriques Pour des préférences issues de distances, les propriétés des préférences dépendent grandement de la manière dont les points sont positionnés dans l'espace sous-jacent. On peut d'ailleurs facilement montrer que toutes les préférences acycliques peuvent être générées en plaçant les n pairs aux sommets d'un simplexe et en le perturbant (en bougeant légèrement les sommets) de manière adéquate. Malgré tout, l'expérience semble montrer que certaines caractéristiques se retrouvent fréquemment. Pour se donner des préférences de proximité typiques et analysables, je propose de considérer les préférences obtenues en considérant les distances prises entre n points au hasard sur un tore unitaire de dimension t ($t \geqslant 1$).

Latences réelles Avoir des mesures réelles pour valider les résultats est bien évidemment nécessaire, même si l'analyse théorique devient alors difficile. Pour $n \leqslant 2500$, j'utilise, pour contrôler la validité de certains résultats géométriques, des sous-ensembles de taille n extraits du jeu de données du projet Meridian [61]. Les valeurs sont les latences (symétriques) entre les nœuds sélectionnés.

Acycliques aléatoires Une dernière classe de préférences est obtenue en attribuant à chaque arête acceptable une valeur aléatoire uniforme comprise entre 0 et 1. J'appelle les préférences qui en résultent *acycliques aléatoires* (ou simplement acycliques quand il n'y a pas d'ambiguïté), car on peut montrer que cette approche produit un échantillonnage uniforme sur l'ensemble des préférences acycliques possibles [32, 7].

4.3 Caractérisation de l'auto-stabilisation

Quelle est la vitesse de convergence ? Cette section va tenter de répondre à cette question, d'un point de vue théorique (§ 4.3.1 et § 4.3.2) et pratique.

4.3.1 Bornes supérieures

Je propose d'abord de donner des bornes supérieures, au sens de la théorie de l'auto-stabilisation. L'initiative considérée dans toute cette section est *meilleur partenaire*. Pour $b = 1$, les bornes sont exactes, linéaires en rounds, et quasi-exactes, exponentielles en initiatives. Je donne ensuite des bornes plus faibles pour $b > 1$.

4.3.1.1 Convergence (rounds)

Pour $b = 1$, le temps maximal de convergence, exprimé en rounds, est donné par le théorème 2.

Théorème 2 ([53]) *Partant de n'importe quelle configuration, une suite adversariale converge en au plus $\lfloor \frac{n}{2} \rfloor$ rounds. Cette borne est atteinte par une suite round-robin.*

Idée de la preuve : Le lemme 1 garantit la stabilisation d'au moins une arête bouillante par round. Comme il y a au plus $\lfloor \frac{n}{2} \rfloor$ arêtes à stabiliser, la borne vient tout de suite. Pour l'atteindre, on peut utiliser des préférences globales, un graphe d'acceptabilité complet, la configuration de départ $C_\emptyset$, et une suite round-robin utilisant le motif *du pire au meilleur*. □

4.3.1.2 Convergence (initiatives)

Si l'on veut maintenant mesurer la convergence en nombre d'initiatives actives, la borne dépend des contraintes de la suite : round-robin ou totalement adversariale.

Suite Round-Robin La borne pour les suites round-robin est donnée par le théorème 3.

Théorème 3 ([53]) *Partant de n'importe quelle configuration, une suite round-robin converge en au plus $\sum_{k=0}^{\lfloor \frac{n}{2} \rfloor - 1} n - (2k+1) \approx \frac{n^2}{4}$ initiatives actives. Cette borne est exacte.*

Idée de la preuve : Comme pour le théorème 3, on utilise le fait qu'au moins une arête bouillante est stabilisée par round. De plus, un pair stable n'est par définition plus jamais actif ; le comptage précis des initiatives possibles donne alors $\sum_{k=0}^{\lfloor \frac{n}{2} \rfloor - 1} n - (2k+1) \approx \frac{n^2}{4}$. Le système utilisé pour le théorème 3 atteint la borne, prouvant qu'elle est exacte. □

Suite adversariale La convergence pour une suite adversariale est décrite par le théorème 4.

Théorème 4 ([53]) *Partant de n'importe quelle configuration, une suite adversariale converge en au plus $2^{n-1} - 1$ initiatives actives. Réciproquement, il existe une suite qui converge en $\Theta(\lambda^n)$ initiatives actives, avec $\lambda \approx 1.6826$, ce qui montre que la borne exacte des suites adversariales se situe entre ces deux valeurs.*

Idée de la preuve : Comme beaucoup d'autres preuves, la borne $2^{n-1} - 1$ est issue du lemme 1. L'astuce est de considérer une arête bouillante et de regarder le moment où elle se stabilise. Avant cet instant, les pairs adjacents ne peuvent prendre l'initiative : on est par hypothèse en *meilleur partenaire*, donc ils se stabiliseraient. Après, ils sont inactifs. On en déduit une récursion sur le nombre de pairs autorisés à prendre une initiative, à partir de laquelle la borne est déduite.

La borne $\Theta(\lambda^n)$ est quant à elle obtenue pour les préférences globales, en acceptabilité complète, à partir d'une suite *le pire d'abord* : à chaque instant, l'initiative est prise par le plus mauvais pair actif (celui avec la plus grande étiquette). Une étude complète du comportement de cette suite d'initiatives donne la borne inférieure. □

Remarque 1 *Dans tous les résultats que je viens d'énoncer, les bornes sont atteintes à partir de préférences globales. De ce point de vue, on peut donc considérer que les préférences globales sont les «pires» possibles pour la convergence, comme la suite va le confirmer. D'une manière plus générale, les préférences globales ont souvent un comportement atypique au sein des préférences acycliques.*

4.3.1.3 Généralisation au b-couplage

Pour $b > 1$, une adaptation du théorème 2 nous donne immédiatement une borne de $\lfloor \frac{nb}{2} \rfloor$ rounds (au moins une arête est stabilisée par round). Mais il est facile de voir que c'est juste une borne supérieure. Par exemple, en prenant $b = n-1$ (c'est le cas limite de l'absence de quotas : tout le monde peut simultanément collaborer avec tout le monde), chaque pair est sûr d'être stabilisé après avoir pris $n-1$ initiatives, ce qui donne une borne de $n-1$ rounds beaucoup plus petite que $\lfloor \frac{n(n-1)}{2} \rfloor$.

L'explication derrière l'imprécision de la borne est que lorsque $b > 1$, une configuration non-stable possède des arêtes chaudes en plus des arêtes bouillantes. Ces arêtes simplement chaudes sont plus dures à dénombrer (toutes les configurations transitoires n'en ont pas), ce qui rend le calcul des bornes plus difficile. On peut quand même donner des bornes plus précises que $\lfloor \frac{nb}{2} \rfloor$ pour les préférences globales (qui, rappelons-le, sont intuitivement les pires préférences possibles).

Théorème 5 ([54]) *Pour les préférences globales, le temps de convergence est borné par $\frac{b}{b+1}n$ rounds si l'acceptabilité est complète, et par n sinon.*

Les simulations (cf § 4.3.3) indiquent que la borne $\frac{b}{b+1}n$ reflète assez bien le comportement réel des préférences globales : les premiers quotas sont ceux qui coûtent le plus cher en terme de convergence. L'acceptabilité complète semble de plus être le pire cas possible, ce qui fait que n est clairement une estimation supérieure (sans parler de $\lfloor \frac{nb}{2} \rfloor$).

Idée de la preuve : Si l'acceptabilité est complète, la configuration stable est faite de cliques de taille $b+1$ ([33]) ; b rounds stabilisent une clique, d'où le résultat. Dans le cas général (acceptabilité inconnue), on peut juste affirmer que b rounds stabilisent b pairs, d'où la borne n. □

4.3.2 Temps moyens de convergence

Intéressons-nous maintenant à la convergence moyenne pour des suites round-robin et Poisson non adversariales. Les résultats présentés ici sont valables pour $b = 1$, en initiative *meilleur partenaire*. Il s'agit de bornes supérieures sur le temps moyen de convergence de certaines classes de systèmes à préférences, temps moyen mesuré en initiatives (1 u.t. $= n$ initiatives).

4.3.2.1 Borne générique

Sous les hypothèses considérées ($b = 1$ et initiative *meilleur partenaire*), la borne suivante est valable pour tout système acyclique :

Théorème 6 ([54]) *Le temps moyen de convergence est borné par $\frac{n}{4}$ pour une suite Poisson, et par $\frac{n+1}{6}$ pour une suite round-robin.*

Idée de la preuve : Comme souvent, le théorème repose sur le lemme 1, à savoir l'existence d'au moins deux pairs bouillants dans toute configuration non-stable. On en déduit un temps moyen entre deux initiatives «bouillantes» de $\frac{1}{2}$ pour une suite Poissonnienne et $\frac{1}{3}(1 + \frac{1}{n})$ pour une suite round-robin. En multipliant par $\frac{n}{2}$ (nombre maximal d'arêtes de la configuration stable), on obtient le résultat. □

Remarque 2 *Pour des préférences globales avec acceptabilité complète, toute configuration instable admet exactement deux pairs bouillants (les deux meilleurs pairs non stabilisés). Ceci suggère que les bornes du théorème 6 doivent être assez précises pour ces systèmes, ce qui est confirmé par les simulations [54]. On retrouve le fait que les préférences globales sont « les pires préférences acycliques possibles ».*

Remarque 3 *La borne exacte $\frac{n+1}{6}$ pour les suites round-robin n'est pas contradictoire avec l'exactitude de la borne $\lfloor \frac{n}{2} \rfloor$ du théorème 2 : l'une est une borne en moyenne sur l'ensemble des suites round-robin, l'autre une borne dans le pire cas.*

4.3.2.2 Préférences globales

Pour les préférences globales, avec un graphe d'acceptabilité Erdös-Rényi $\mathcal{G}(n, p)$, il existe des bornes plus fines :

Théorème 7 ([54]) *En préférences globales, si $G = \mathcal{G}(n, p)$, le degré moyen étant $d = p(n - 1)$, le temps moyen de convergence est en $O(d+\log(n))$ pour une suite Poissonnienne et en $O(d)$ pour une suite round-robin.*

Idée de la preuve : Le cœur de la preuve consiste à *compter* les pairs bouillants dans les configurations non-stables. Par des techniques combinatoires, on montre qu'il y en a de l'ordre de $\frac{n}{d}$. On en déduit ensuite un temps moyen entre deux initiatives « bouillantes » de l'ordre de $\frac{d}{n}$, ce qui donne le comportement commun en $O(d)$. Enfin, il faut porter une attention particulière à la finalisation du processus de convergence, lorsque la quasi-totalité des arêtes sont déjà stabilisées. Dans cette *fin de partie*, les nœuds non-stables sont quasiment tous bouillants. Une suite round-robin va alors achever la stabilisation en $O(1)$ (1 unité de temps stabilise tous les nœuds bouillants du moment), tandis qu'une suite Poissonnienne va avoir besoin de $O(\log(n))$ u.t. (problème de boules et d'urnes). □

4.3.2.3 Préférences acycliques

Pour des préférences acycliques (aléatoires), on a un résultat similaire :

Théorème 8 ([54]) *Pour des préférences acycliques aléatoires, si $G = \mathcal{G}(n, p)$, le degré moyen étant $d = p(n - 1)$, le temps moyen de convergence est en $O(\log(d) + \log(n))$ pour une suite Poissonnienne et en $O(\log(d))$ pour une suite round-robin.*

Idée de la preuve : Un calcul peut montrer que pour des préférences acycliques aléatoires, environ la moitié des pairs non stables sont bouillants. Cela permet de montrer que la fin de partie est atteinte au bout de $O(\log(d))$ unités de temps. Si la suite est Poisson, il faut alors rajouter $O(\log(n))$ u.t. supplémentaires. □

Remarque 4 *Comme $d < n$, certains voudront que je simplifie la borne $O(\log(d) + \log(n))$ en $O(log(n))$, et ils auront raison. Cependant, je reste sur ma notation, parce qu'en bon physicien du pays de Pagnol, j'estime que* le grand O du $\log(d)$ compte plus que le grand O du $\log(n)$ *(cf §4.3.3.1).*

4.3.2.4 Autres préférences acycliques

Pour d'autres types de préférences acycliques, la clé pour estimer le temps de convergence moyen est d'arriver à estimer le nombre de pairs bouillants. Pour des préférences réelles comme celles issues de Meridian, c'est un exercice difficile. De même pour les préférences géométriques, où il existe de fortes corrélations entre les préférences de pairs proches. Notons que pour ce dernier cas, ne pas tenir compte de ces corrélations permet de se ramener au cas des préférences acycliques aléatoires, et cela marche assez bien en pratique (cette technique sera d'ailleurs abondamment utilisée dans la section suivante).

Dans *self-stabilization in preference-based systems* [54], je propose de regarder la distribution de la « valeur » des pairs pour déterminer si la convergence d'un système est plus proche de celle des préférences globales ou de celle des préférences acycliques. Si certains pairs sont bien classés par beaucoup de leurs voisins (ce sont donc des « bons » pairs), la convergence devrait ressembler à celle des préférences globales ($O(d)$ ou $O(d + \log(n))$). Par contre, si aucun bon pair ne se dégage (les pairs ont à peu près tous la même valeur), la convergence devrait être de type acyclique aléatoire ($O(\log(d))$ ou $O(\log(d) + \log(n))$).

Je suggère d'utiliser cette règle empirique (tout réseau à préférences acycliques a un comportement entre celui des préférences acycliques aléatoires et celui des préférences globales, le curseur étant positionné par la distribution des valeurs) pour tous les types de préférences acycliques, par exemple pour des combinaisons linéaires de préférences. Et aussi pour des graphes d'acceptabilité et des quotas quelconques. Hélas, je n'ai pas de preuve théorique à apporter. Heureusement, pour tout ce que je ne peux pas prouver, il y a les simulations.

4.3.3 Simulations

Au-delà des résultats purement théoriques, je propose les simulations pour compléter la compréhension de la convergence des préférences acycliques. C'est une approche que j'affectionne tout particulièrement quand un système qui dépend de quelques paramètres ne peut pas être analysé entièrement : faire varier les paramètres un par un afin de se forger une intuition sur l'influence de chacun de ces paramètres [8]. Tous les résultats présentés ici viennent de l'article *self-stabilization in preference-based systems* [54], dans lequel j'invite le lecteur soucieux des détails techniques à se plonger.

4.3.3.1 Initiative *meilleur partenaire*

Pour commencer, regardons la convergence sous l'initiative *meilleur partenaire*, qui a été abondamment étudiée au début de cette section. Des quotas supérieurs à 1 seront utilisés pour voir comment les résultats vus pour $b = 1$ s'appliquent pour $b \geqslant 1$.

Taille du voisinage acceptable Afin de comprendre l'influence du nombre de voisins, on fixe n et l'on fait varier $d = p(n-1)$ (G est toujours supposé Erdös-Rényi). On observe deux principaux comportements :

- Pour les préférences globales, le temps de convergence croît de manière linéaire ; la suite round-robin est plus rapide que la suite Poissonnienne sauf lorsque d est proche de n ;

8. Voir par exemple l'article *Epidemic live streaming : optimal performance trade-offs* [11].

- pour les autres préférences étudiées, le temps de convergence croît d'une manière qui semble logarithmique ; la suite round-robin est toujours plus rapide que la suite Poissonnienne ; les deux préférences les plus rapides (acycliques aléatoires et géométriques) sont quasi-impossibles à différencier.

Taille du système Considérons maintenant des systèmes où l'on fait varier n, à d constant. Les résultats sont les suivants :

- Le temps de convergence augmente avec n pour les suites Poissonniennes. C'est l'effet *fin de partie* en $O(\log(n))$;
- pour les suites round-robin, le temps de convergence augmente sensiblement moins. Il y a même une légère décroissance au niveau des préférences globales pour n proche de d ;
- excepté pour le cas ci-dessus (préférences globales, n proche de d), les séquences Poissonniennes sont plus longues que les séquences round-robin ;
- comme pour l'expérience précédente, les préférences acycliques et géométriques convergent le plus vite, suivies des préférences Meridian et enfin, relativement loin, des préférences globales.

La principale leçon à retenir de ces simulations reste que à d fixé, n a relativement peu d'effet sur le temps de convergence. «Le grand O du d est plus important que le grand O du n».

Quotas de collaborations Comme pour les autres paramètres, le rôle des quotas dans le temps de convergence dépend du type de préférences :

- Pour les préférences globales, le temps de convergence croît d'abord vite puis lentement (les premiers quotas coûtent le plus cher) ; les suites Poissonniennes sont légèrement plus rapides ;
- pour les autres préférences, on note une croissance linéaire ou quasi-linéaire, les suites round-robin étant légèrement plus rapides.

Récapitulatif En résumé, les préférences globales se distinguent par un temps de convergence linéaire par rapport au degré moyen, ce qui fait de d le principal paramètre à considérer pour la convergence. Les quotas b ont une influence plus faible, surtout une fois les premières valeurs passées (plus il y a de connexions, moins une nouvelle connexion prend de temps).

Pour les autres préférences, où il n'y a pas vraiment de « bons » pairs, c'est quasiment l'inverse : la convergence est proportionnelle aux quotas (chaque connexion coûte le même prix en temps que la précédente), mais seulement logarithmique par rapport à la taille du voisinage.

4.3.3.2 Initiatives aléatoires et hybrides

La plupart des travaux que j'ai effectués jusqu'à présent portent sur l'initiative *meilleur partenaire*, car c'est une initiative déterministe, ce qui simplifie son étude. Pour conclure cette discussion sur la convergence, j'aimerais évoquer un peu l'initiative *aléatoire*, qui sera je pense un sujet très intéressant à étudier pour de futurs travaux.

Une initiative *aléatoire* est beaucoup moins coûteuse en messages de contrôle qu'une initiative *meilleur partenaire*, car il est inutile de connaître les rangs et partenaires de ses voisins pour la pratiquer. En contre-partie, elle augmente le temps de convergence : alors qu'un pair bouillant stabilise nécessairement une arête en prenant une initiative *meilleur partenaire*, ce n'est plus le cas en initiative aléatoire.

Cependant, il n'y a pas que le temps requis pour une convergence complète qui compte. Le temps nécessaire pour atteindre *d'assez bonnes* configurations (pour un certain sens) est au moins aussi important.

Par exemple, il est possible de mesurer une satisfaction des pairs dans une configuration donnée (cf [54] pour une description détaillée) et d'observer comment cette satisfaction évolue au cours du processus de convergence. Pour des préférences non-globales (géométriques, aléatoires ou Meridian) et des mêmes conditions initiales, l'initiative *meilleur partenaire* produit à chaque instant une satisfaction meilleure que l'initiative aléatoire. En revanche, la situation est beaucoup moins tranchée si les préférences sont globales : d'un côté, l'initiative aléatoire n'a besoin que de quelques u.t. pour créer une forte satisfaction, mais elle peine ensuite à atteindre la satisfaction stable ; de l'autre, l'initiative *meilleur partenaire* a une croissance initiale beaucoup plus faible et ne dépasse que tardivement la satisfaction de l'initiative aléatoire, mais elle converge ensuite rapidement.

L'intérêt de cet exemple est de montrer qu'en plus d'être moins coûteuse, il y a des cas (les préférences globales en l'occurrence) où l'initiative aléatoire peut s'avérer plus efficace : avoir une convergence initiale très forte, même si la convergence finale est lente, peut être très avantageux pour un système soumis à forte agitation (et où il est donc illusoire d'essayer d'atteindre une convergence complète). Je me permets au passage de remarquer que le protocole BitTorrent, qui peut de manière très grossière être assimilé à un système à préférences globales soumis à agitation, utilise l'initiative aléatoire, plus connue ici sous le nom d'*optimistic unchoking*[9].

Je propose l'interprétation suivante (toujours tirée de [54]) pour expliquer cette différence dans les convergences, pour les préférences globales : en initiative *meilleur partenaire*, tout le monde essaie les «bons» pairs. En particulier, les mauvais pairs vont continuellement essayer les meilleurs pairs non-stabilisés. Ces derniers se stabilisent très vite (ils sont chauds), ce qui casse leurs mauvaises connexions. Le résultat est une sorte de front de convergence (ou front de saturation) qui évolue temporellement des meilleurs aux pires pairs, les pairs étant stabilisés après le passage du front, mais n'ayant peu ou pas de connexions avant. En particulier, un mauvais pair n'arrive pas à conserver ses partenaires avant la convergence complète du système. À l'inverse, l'initiative aléatoire ne crée pas de front de saturation, mais une sorte de convergence uniforme de la satisfaction : les mauvais pairs peuvent choisir de mauvais partenaires, ce qui fait que leur connexions intermédiaires durent plus longtemps.

Les avantages des initiatives *aléatoire* et *meilleur partenaire* peuvent se combiner dans des initiatives hybrides. Je propose par exemple une initiative où chaque pair fonctionne en initiative aléatoire en dessous d'un certain nombre de partenaires, et bascule en *meilleur partenaire* au-delà. Les bons pairs convergent quasiment aussi vite qu'en *meilleur partenaire*, suivant un front de saturation, mais les pairs situés devant le front ont quand même une bonne satisfaction due à la composante aléatoire qui se déclenche dès qu'il n'y a pas assez de partenaires. Au final, l'initiative hybride se comporte bien quelles que soient les préférences acycliques (pas seulement pour les préférences globales donc), aussi bien en convergence initiale que complète, ce qui en fait un choix intéressant, en particulier pour des systèmes acycliques où la nature exacte des préférences est inconnue ou variable.

4.4 Les configurations stables

Avant de conclure ce chapitre sur les réseaux à préférences acycliques, je propose de répondre, au moins partiellement et pour les graphes d'acceptabilité Erdös-Rényi, à la deuxième question majeure des systèmes à préférences acycliques : *quelles sont les propriétés de la configuration stable ?*

Pour $b = 1$, il est possible d'étudier la distribution des partenaires grâce à une approche (simplifiée) de type *champ moyen*. Pour les préférences globales, on peut prouver l'existence d'une limite fluide avec solution explicite, qui montre en particulier que la distribution de la valeur du partenaire d'un pair est centrée autour de la valeur de ce pair, avec une décroissance exponentielle : c'est l'effet de *stratification* [33]. Pour des préférences acycliques aléatoires ou géométriques, la solution de la limite fluide décroît en loi de puissance. Bien que je ne sois pas en mesure de prouver l'approche *champ moyen*, je propose de valider cette approche en la comparant avec la solution exacte (pour les préférences globales) ou à l'aide de simulations.

Enfin, je propose d'étendre les résultats à $b \geqslant 1$. Les équations fluides ne semblent alors plus admettre de solution explicite, mais on observe le même comportement asymptotique que pour $b = 1$ (exponentiel pour les préférences globales, en loi de puissance sinon). Une conséquence inattendue est que pour les préférences géométriques, la configuration stable est un petit-monde si les quotas sont suffisants.

La plupart des résultats présentés ici sont rassemblés dans l'article *The Stable Configuration*

9. Je ne pense bien sûr pas que Bram Cohen ait introduit l'*optimistic unchoking* dans cette perspective : c'est avant tout une technique simple et efficace pour résoudre à la fois le problème de l'évaluation de nouveaux voisins et celui du *bootstrap*. Mais cela reste une bonne nouvelle de découvrir que la modélisation par préférences acycliques pointe vers la même technique que celle validée empiriquement.

in Acyclic Preference-Based Systems [57], dont une version étendue est disponible sous forme de rapport de recherche [56].

4.4.1 Notation spécifique

Comme je propose d'étudier en détail la configuration stable dans cette section, quelques notations supplémentaires sont nécessaires, pour décrire en particulier la distribution des partenaires. Si i et j sont acceptables, $r_i(j)$ désigne le rang acceptable de j selon i (1 étant le meilleur rang). r_i est le classement acceptable de i. Si i a plus de k voisins (acceptables), $r_i^{-1}(k)$ est le $k^{ième}$ meilleur voisin acceptable de i. De même, pour tout $j \neq i$, $R_i(j)$ désigne le rang de j dans le graphe complet, sans tenir compte de la notion d'acceptabilité [10]. R_i est le classement *complet* de i. Pour $K < n$, $R_i^{-1}(K)$ est le $K^{ième}$ meilleur voisin, acceptable ou non, de i.

Dans tout ce qui suit, D désigne la distribution du, ou des, partenaires stables. Afin d'alléger l'écriture, je propose d'utiliser une notation lâche, où le sens de D est spécifié à l'aide d'indices et d'exposants à chaque fois que cela est nécessaire. Ainsi, $D_{R_i}(K)$ indique la probabilité que i ait son pair de rang complet K comme partenaire ; $D_{n,d}(i,j)$ est la probabilité que i et j soient partenaires s'il y a n pairs avec une acceptabilité Erdös-Rényi de degré moyen $d = p(n-1)$; si $c \leqslant b(i,)$, $D_{r_i,c}(k)$ est la probabilité que le c^{th} meilleur partenaire stable de i ait un rang acceptable k...

La distribution cumulative complémentaire (*Complementary Cumulative Distribution Function*, ou CCDF) de D est notée S, et les versions normalisées de D et S sont respectivement notées $\mathcal{D}$ et $\mathcal{S}$.

4.4.2 Équations acycliques

Pour le cas du couplage simple ($b = 1$), je propose une méthode générique pour décrire le rang complet de $C(i)$, s'il existe, dans la configuration stable C. La généralisation au b-couplage sera faite en § 4.4.5.

4.4.2.1 Équation exacte

Soit $D_{R_i}(K)$ la probabilité que $R_i(C(i)) = K$, c'est-à-dire que le partenaire de i, s'il existe, ait le rang complet K. La CCDF de D est définie par $S_{R_i}(K) := 1 - \sum_{L=1}^{K-1} D_{R_i}(K)$, i.e. la probabilité que i ait un partenaire de rang complet plus grand ou égal à K ($R_i(C(i)) \geqslant K$) ou n'ait pas de partenaire du tout (notation courte pour la disjonction des deux évènements : $R_i(C(i)) \not< K$). En s'inspirant de l'approche utilisée dans [33], je propose une équation exacte décrivant D_{R_i} avant de donner une version simplifiée issue d'une approche de type *champ moyen*.

Pour exprimer $D_{R_i}(K)$, on peut observer que pour que i soit stable avec son pair de rang complet K, noté $j := R_i^{-1}(K)$, il faut (et il suffit de) vérifier les trois conditions suivantes :

- $\{i, j\}$ doit être une arête acceptable, ce qui se produit avec probabilité p si G est un graphe Erdös-Rényi $\mathcal{G}(n,p)$;
- i ne doit pas avoir de meilleur partenaire stable que j ($R_i(C(i)) \not< K$) ;
- j ne doit pas avoir de meilleur partenaire stable que i ($R_j(C(j)) \not< R_j(i)$).

Cela mène à l'équation exacte suivante :

$$D_{R_i}(K) = pS_{R_i}(K)\mathbb{P}(R_j(C(j)) \not< R_j(i) | R_i(C(i)) \not< K) \tag{4.1}$$

10. Je suppose l'existence d'un prolongement naturel de la matrice des valeurs aux paires non acceptables. Ce prolongement est immédiat pour les préférences globales (la valeur intrinsèque) et géométriques (la distance). Pour les préférences acycliques aléatoires, il s'obtient en attribuant des valeurs aléatoires « fantômes » aux arêtes non-acceptables.

4.4.2.2 Équation approchée

La principale difficulté de l'équation (4.1) est la probabilité conditionnelle, délicate à manier à cause des corrélations qui peuvent exister entre $R_j(C(j)) \not< R_j(i)$ et $R_i(C(i)) \not< K$. La solution est de tenir ces corrélations pour négligeables :

Approximation 1 *Les évènements « i n'est pas avec meilleur que j » et « j n'est pas avec meilleur que i » sont indépendants.*

Cette approximation, que j'appelle parfois un peu abusivement « hypothèse de champ moyen », est raisonnable pour p assez petit [11]. L'équation (4.1) peut alors se simplifier en

$$D_{R_i}(K) = pS_{R_i}(K)S_{R_j}(R_j(i)). \tag{4.2}$$

Pour aller plus loin, il faut prendre en compte le type de préférences.

4.4.3 Préférences globales

Comme les préférences globales sont caractérisées par un ordre total sur les pairs, on peut se passer de la matrice de valeurs m en étiquetant les pairs de 1 à n, 1 étant le meilleur. On considère ainsi directement la probabilité $D(i,j)$ que i et j soient partenaires. En remarquant que le rang complet de j pour i est j si $j < i$ et $j-1$ si $j > i$ (un pair ne se classe pas lui-même), on obtient la relation entre D et D_R :

$$D(i,j) = \begin{cases} D_{R_i}(j) \text{ si } j < i, \\ 0 \text{ si } j = i \text{ (non-réflexivité)}, \\ D_{R_i}(j-1) \text{ si } j > i. \end{cases} \tag{4.3}$$

En posant $S(i,j) := 1 - \sum_{k=1}^{j-1} D(i,k)$, on obtient la déclinaison de l'équation 4.2 pour les préférences globales :

$$D(i,j) = \begin{cases} 0 \text{ si } i = j, \\ pS(i,j)S(j,i) \text{ sinon.} \end{cases} \tag{4.4}$$

Cette équation [12], qui se résout numériquement par une double itération, donne une très bonne approximation de la distribution empirique [33].

4.4.3.1 Normalisation

Une limite fluide explicite vers laquelle les distributions discrètes convergent est un atout certain pour la description de la configuration stable. Cela permet en effet de donner une description complète, immédiate et globale de la distribution pour tous n et p, ce qui n'est pas le cas de l'équation (4.4).

Afin d'obtenir cette limite, il faut pouvoir comparer des distributions pour des valeurs arbitraires de n. Il est donc nécessaire de normaliser D. Une manière assez simple de le faire consiste à représenter chaque pair i par un rang normalisé α, avec $0 \leqslant \alpha < 1$. Plus précisément, on associe à chaque i le réel $\alpha(i) = \frac{i-1}{n}$, et réciproquement à chaque réel positif α l'entier $i(\alpha) = \lfloor n\alpha \rfloor + 1$. La version normalisée de D, notée $\mathcal{D}$, est alors définie par

$$\mathcal{D}_n(\alpha, \beta) = nD(\lfloor n\alpha \rfloor + 1, \lfloor n\beta \rfloor + 1). \tag{4.5}$$

$\mathcal{D}_n$ est une fonction en escalier (à deux variables), qui prend les valeurs $(nD(i,j))$. Le facteur n permet d'exprimer simplement $D(i,j)$ comme une intégrale de $\mathcal{D}$:

$$D(i,j) = \int_{\frac{j-1}{n}}^{\frac{j}{n}} \mathcal{D}_n(\frac{i-1}{n}, x)\,\mathrm{d}x = \int_{\frac{i-1}{n}}^{\frac{i}{n}} \mathcal{D}_n(x, \frac{j-1}{n})\,\mathrm{d}x.$$

11. Quelques exemples simples semblent indiquer une erreur en p^3, confirmée par les simulations [33].
12. Pour la petite histoire, l'équation (4.4) a été proposée par Julien Reynier sur une idée originale de Fabien de Montgolfier. Chronologiquement elle est à l'origine de l'équation générale 4.2, et non l'inverse.

La distribution cumulative complémentaire normalisée est définie par

$$\mathcal{S}_n(\alpha,\beta) = 1 - \int_0^\beta \mathcal{D}_n(\alpha,x)\,\mathrm{d}x, \tag{4.6}$$

et la relation entre S et $\mathcal{S}$ est

$$S(i,j) = \mathcal{S}(\frac{i-1}{n},\frac{j-1}{n}). \tag{4.7}$$

4.4.3.2 Convergence des distributions normalisées

Si le degré moyen reste constant, j'ai montré l'existence d'une limite continue des distributions $\mathcal{D}$. La première étape est de régler le problème d'une discontinuité intrinsèque le long de la diagonale principale ($\alpha \approx \beta$), due au fait que $D(i,i) = 0$. C'est un problème mineur car cette discontinuité est juste là pour rappeler la non-réflexivité du couplage. Il se résout en introduisant une fonction $\mathcal{D}$ plus « continue », obtenue en prolongeant $\mathcal{D}$ sur la diagonale principale à l'aide de l'équation (4.4) :

$$\tilde{\mathcal{D}}(\alpha,\beta) = \begin{cases} \mathcal{D}(\alpha,\beta) \text{ si } \lfloor n\alpha\rfloor \neq \lfloor n\beta\rfloor, \\ np(S(\lfloor n\alpha\rfloor+1,\lfloor n\alpha\rfloor+1))^2 \text{ sinon.} \end{cases}$$

La limite fluide, à degré constant, des fonctions $\tilde{\mathcal{D}}$ est alors donnée par le théorème qui suit :

Théorème 9 *Soit $d > 0$ une constante. Si $n \to \infty$, avec $p = \frac{d}{n}$, les fonctions $\tilde{\mathcal{D}}_{n,d}$ convergent uniformément vers*

$$\mathcal{D}_\infty(\alpha,\beta) = \frac{de^{d(|\beta-\alpha|)}}{(1-e^{-d\min(\alpha,\beta)}+e^{d|\beta-\alpha|})^2}. \tag{4.8}$$

Ce résultat montre qu'asymptotiquement, la distribution des partenaires ne dépend que du degré moyen de G (supposé Erdös-Rényi). Cela permet de décrire quantitativement l'effet de *stratification* [33] : à α fixé, la distribution du partenaire stable de α décroît exponentiellement en $|\beta-\alpha|$, avec intensité d ($\mathcal{D}_\infty(\alpha,\beta) \approx de^{-d|\beta-\alpha|}$ pour $d|\beta-\alpha|$ assez grand). Autrement dit, un pair de rang normalisé α tend à avoir pour partenaire un pair de même rang, à plus ou moins $\frac{1}{d}$.
Idée de la preuve : La preuve du théorème 9 comporte quatre étapes [56] :
- montrer que les fonctions $\tilde{\mathcal{D}}_{n,d}$ sont uniformément Cauchy sur $[0,1]^2$;
- utiliser la convergence de Cauchy pour montrer que $\mathcal{S}_N$ et $\tilde{\mathcal{D}_N}$ admettent des limites $\mathcal{S}_\infty$ et $\mathcal{D}_\infty$;
- donner une EDP vérifiée par $\mathcal{S}_\infty$;
- résoudre l'EDP [13], et déduire $\mathcal{D}_\infty$ de la solution.

□

L'existence de cette limite fluide avait été proposée comme conjecture dès [33], et prouvée pour le cas $\alpha = 0$, mais ce n'est que plus tard que la preuve et l'expression complètes ont été trouvées [56].

Le théorème 9 permet d'obtenir trois corollaires immédiats qui complètent la compréhension de la configuration stable.

Corollaire 1 *En considérant $\mathcal{S}_\infty(\alpha,1)$, la probabilité qu'un pair de rang normalisé α soit célibataire dans la configuration stable est $\frac{1}{1+e^{-d\alpha}(e^{-d}-1)}$*

Corollaire 2 *Pour $i \neq j$ (cas discret), une bonne approximation de $D(i,j)$ est*

$$D(i,j) \approx \frac{pe^{p(|j-i|)}}{(1-e^{-p\min(i,j)}+e^{p|j-i|})^2}. \tag{4.9}$$

13. Pour le lecteur avide d'équations « exotiques », il s'agit de résoudre $\partial_y\mathcal{S}_\infty(\alpha,\beta) = -d\mathcal{S}_\infty(\alpha,\beta)\mathcal{S}_\infty(\beta,\alpha)$, avec la condition initiale $\mathcal{S}_\infty(\alpha,0) = 1$. Bien que d'apparence innocente, la résolution de cette équation non-locale m'a donné pas mal de fil à retordre, alors même que je connaissais par ailleurs sa solution. Je remercie en passant François Baccelli qui m'a aiguillé sur la piste de la dé-non-localisation.

Corollaire 3 *Soit une suite de distributions normalisées $\mathcal{D}_{n,d}$ à degré croissant non borné ($d \xrightarrow[n\to\infty]{} \infty$). On a*

$$\forall \alpha \in [0,1[, \mathcal{D}_{n,d}(\alpha,.) \xrightarrow[n\to\infty]{*} \delta_\alpha.$$

Ce dernier corollaire généralise un théorème proposé dans [33], qui montrait l'existence d'une limite faible de Dirac dans le cas d'une suite de distributions à $p = \frac{d}{n-1}$ constant. Il signifie que dès que le degré tend vers l'infini (par exemple en $O(\log(n))$), alors asymptotiquement, un pair a pour partenaire un pair de même rang normalisé (pas de déviation), et la probabilité de célibat devient nulle.

4.4.3.3 Validation

Pour valider les résultats précédents, il faut comparer des résultats de simulations à l'équation (4.4) (récurrence sous l'approximation d'indépendance), puis à l'équation (4.9). C'est ce qui a été fait [33, 56].

Comme indiqué précédemment, on observe pour l'équation (4.4) une très bonne précision, de l'ordre de p^3.

Pour la limite fluide, mis à part le prolongement continu sur la diagonale principale, la précision est également très bonne, sauf pour des valeurs de n faibles et de d élevées. Ces observations sont consistantes avec la preuve complète du théorème 9 [56], qui montre une convergence en $O(\frac{d^2}{n}e^{8d})$. Elles seraient même plutôt en faveur d'une convergence en $O(\frac{d^2}{n})$, qui est d'après moi la vraie borne, même si cela reste à prouver [14].

4.4.3.4 Résolution exacte

Dans la situation que je viens de proposer ($b = 1$, préférences globales, acceptabilité $\mathcal{G}(n,p)$), il est en fait inutile de recourir à l'approximation d'indépendance, car il existe une récurrence qui permet d'obtenir la distribution exacte [56].

J'ai pu obtenir cette récurrence en conditionnant le fait que i et j soient partenaires selon le rang du partenaire du pair 1 (plus petit que i, entre i et j, plus grand que j ou pas de partenaire).

Tout comme pour la formule approchée, la formule exacte admet une limite fluide, qui obéit à une certaine EDP. Bien que les deux EDP (exacte et approchée) soient extrêmement différentes [15], elles donnent la même solution, à savoir (4.8).

Le fait que dans ce cas particulier, la récurrence donnée par l'approximation d'indépendance ait exactement la même limite fluide que la récurrence exacte est un argument de poids (à défaut d'être rigoureux) pour justifier l'utilisation de cette approximation dans les autres cas. Car un inconvénient majeur de la récurrence exacte est que la « ruse » utilisée ne s'applique pas pour les autres préférences, ni pour $b > 1$. Si jolie soit cette récurrence, l'hypothèse d'indépendance reste donc incontournable si l'on veut généraliser les résultats.

4.4.4 Préférences acycliques et géométriques

Je propose maintenant de s'intéresser aux préférences acycliques (aléatoires) et géométriques. Pour ces préférences, on peut remarquer que les pairs sont non différenciés. Sur l'ensemble des réalisations possibles, on peut donc considérer que tous les pairs suivent la même distribution de partenaire : $D_{R_i}(K)$ est indépendant de i, et peut donc être noté $D_R(K)$.

14. Le facteur e^{8d} vient de l'utilisation du lemme de Grönwall [9], qui amplifie une erreur de quantification incompressible en $O(\frac{d^2}{n})$. Mais en pratique, l'équation (4.4), qui sert de base à la construction des distributions, est auto-stabilisante (une erreur dans un sens à un moment donné est compensée dans l'autre sens à l'itération suivante), ce que le lemme de Grönwall ne permet pas de prendre en compte.

15. L'EDP exacte est relativement classique, et se résout par la méthode des *caractéristiques*, alors qu'il a fallu résoudre l'EDP approchée, décrite en note page 29, de manière moins conventionnelle.

Comme pour les préférences globales, il s'agit de trouver la distribution du rang complet en simplifiant l'équation (4.1) avant de la résoudre. Je donne aussi rapidement la distribution des distances et des éléments de résolution de la distribution du rang relatif.

4.4.4.1 Distribution du rang complet

Pour les préférences géométriques ou aléatoires, l'approximation 1 n'est pas suffisante, c'est pourquoi je propose une approximation supplémentaire :

Approximation 2 *Le rang complet est symétrique :* $R_i(j) = R_j(i)$.

Je suppose donc que $R_i(j)$ n'est pas une si mauvaise approximation de $R_j(i)$ pour les préférences considérées. Cela donne une équation très simple pour $D_R(K)$:

$$D_R(K) = pS_R^2(K), \text{ avec } S_R(K) = 1 - \sum_{L=1}^{K-1} D_R(L). \tag{4.10}$$

S_R peut donc être obtenue par une récurrence simple :

$$S_R(K) = \begin{cases} 1 \text{ si } K = 1, \\ S_R(K-1) - pS_R^2(K-1) \text{ sinon.} \end{cases} \tag{4.11}$$

Et D_R est directement donnée par $D_R(K) = S_R(K) - S_R(K+1)$.

Limite fluide Comme pour les préférences globales, D_R peut être normalisée. Pour $0 \leqslant \alpha < 1$, on pose $\mathcal{D}_R(\alpha) := (n-1)D_R(\lfloor (n-1)\alpha \rfloor + 1)$. Le facteur de normalisation est maintenant $n-1$ car c'est la valeur maximale de K (n était celle de i et j en §4.4.3). D_R s'exprime comme une intégrale de $\mathcal{D}_R$:

$$D_R(K) = \int_{\frac{K-1}{n-1}}^{\frac{K}{n-1}} \mathcal{D}_R(x)\,\mathrm{d}x.$$

$\mathcal{S}_R$ est alors naturellement définie par :

$$\mathcal{S}_R(\alpha) = 1 - \int_0^{\alpha} \mathcal{D}_R(x)\,\mathrm{d}x.$$

Théorème 10 *Soit $d > 0$ une constante. Si $n \to \infty$, avec $p = \frac{d}{n}$, les fonctions $\mathcal{S}_R$ convergent uniformément vers*

$$\mathcal{S}_\infty(\alpha) = \frac{1}{d\alpha + 1}. \tag{4.12}$$

En particulier, la probabilité qu'un pair soit célibataire dans la configuration stable est asymptotiquement $\mathcal{S}_R(1) = \frac{1}{d+1}$, et une bonne approximation de $S_R(K)$ est

$$S_R(K) = \frac{1}{p(K-1)+1}. \tag{4.13}$$

Idée de la preuve : La preuve est la même que pour le théorème 9, en plus facile. Il faut d'abord montrer que les $\mathcal{D}_R$ sont uniformément Cauchy (ce qui est plus facile ici car il n'y a qu'une seule variable et pas besoin d'introduire un prolongement sur la diagonale principale). On a alors la convergence uniforme vers une fonction continue $\mathcal{S}_\infty$. On déduit alors de (4.11) une équation différentielle ordinaire vérifiée par $\mathcal{S}_\infty$:

$$-\dot{\mathcal{S}}_\infty(\alpha) = d\mathcal{S}_\infty^2(\alpha), \tag{4.14}$$

avec la condition initiale $\mathcal{S}_\infty(0) = 1$. La solution de (4.14) est (4.12), ce qui achève la preuve. □

Validation Compte tenu des approximations faites, il est nécessaire de vérifier sur quelques exemples la précision de l'équation (4.13). J'ai donc testé sa validité pour quelques valeurs de n et de p [56].

Pour $p = 1$, on ne peut pas vraiment dire que l'hypothèse d'indépendance soit vérifiée. Les simulations montrent que les distributions sont affectées par le type particulier de préférences, et la limite fluide n'est pas très précise, en particulier aux limites (K proche de 1 ou de n). On peut entre autres remarquer que la probabilité de célibat $\frac{1}{n}$ donnée par (4.13) est clairement sur-estimée si n est pair (puisqu'elle est alors nulle en réalité), mais exacte pour n impair. La limite fluide réussit quand même à donner le comportement en $\frac{1}{K}$ commun à toutes les préférences considérées. De ce point de vue, elle est meilleure que l'équation récursive (4.11) (dont elle dérive pourtant), qui donne $S_R(K) = \delta_K^1$ pour $p = 1$.

Au fur et à mesure que p diminue, les courbes empiriques et la limite donnée par le théorème 10 se rejoignent très vite. Dès $p = \frac{1}{10}$, elles sont presque indistinguables, ce qui valide l'utilisation de la limite fluide comme approximation de la distribution du rang complet.

4.4.4.2 Distribution des distances

Il peut être intéressant de considérer d'autres distributions que celle du rang complet. Par exemple, pour des préférences géométriques, la distribution des distances entre partenaires stables peut être intéressante si les performances d'un couplage sont liées aux distances. Or, cette distribution peut être déduite de celle du rang complet :

Théorème 11 *Soit $S_X(x)$ la probabilité qu'un pair n'ait pas de partenaire stable à distance inférieure à x dans le tore unitaire de dimension t. Soit B_t le volume d'une boule de rayon x dans ce tore. Sous la limite fluide, on a*

$$S_X(x) = \frac{1}{dB_t(x) + 1}, \tag{4.15}$$

Idée de la preuve : Une boule de rayon x, centrée sur un pair quelconque, contient à peu près $nB_t(x)$ pairs puisqu'elle occupe une proportion $B_t(x)$ du tore (qui est supposé unitaire). Le pair le plus éloigné à l'intérieur de cette boule devrait donc avoir un rang complet d'à peu près $nB_t(x)$ pour le pair central, tout en étant à une distance d'environ x. On obtient ainsi la relation $S_X(x) = S_R(nB_t(x))$, et il ne reste plus qu'à utiliser l'équation (4.13) pour conclure. □

La fonction $B_t(x)$ dépend de la dimension t et de la norme utilisée. Pour la norme infinie, on a tout simplement $B_t(x) = \min((2x)^t, 1)$. L'écriture des autres normes peut être un peu plus compliquée à cause de possibles effets de recouvrement sur les bords. Remarquons que si j'avais choisi $\mathbb{R}^t$ (avec densité homogène) au lieu du tore unitaire, $B_t(x)$ aurait juste été la taille d'une boule de rayon x, sans effets de bord.

Comme à l'accoutumée, la précision de l'équation (4.15) a été vérifiée empiriquement, et le résultat est qu'elle a quasiment la même zone de validité que l'équation (4.13) [56].

4.4.4.3 Distribution du rang relatif

Pour les préférences acycliques aléatoires et géométriques, je me suis également intéressé à la distribution du rang relatif, donnée par la fonction $D_r(k)$ et sa distribution cumulée complémentaire $S_r(k) := 1 - \sum_{l=1}^{k-1} D_r(k)$.

J'ai donc essayé d'adapter la méthode utilisée pour le rang complet. Les conditions à remplir pour que le partenaire stable de i soit son $k^{ième}$ voisin acceptable $j = r_i^{-1}(k)$ sont les suivantes :

- i doit avoir au moins k voisins (si le $k^{ième}$ voisin existe, il est acceptable par définition),
- i ne doit pas être avec un pair meilleur que j,
- j ne doit pas être avec un pair meilleur que i.

En adaptant les approximations d'indépendance et de symétrie au rang relatif, on obtient alors la formule de récurrence suivante [56] :

$$D_r(k) = S_r(k)\frac{1 - I_{1-p}(n-k+1, k)}{k+1}, \tag{4.16}$$

où I_x est la fonction bêta incomplète normalisée.

Hélas, l'équation (4.16) est beaucoup moins précise que ne l'est l'équation (4.13) pour le rang complet, en particulier pour $D_r(1)$ [56]. La raison en est que les corrélations sont beaucoup plus présentes lorsque c'est le rang acceptable que l'on considère (en particulier si l'on considère le voisinage proche).

À titre d'exercice, je me suis donc demandé s'il était possible d'affiner l'estimation du rang relatif, et j'ai partiellement réussi : pour $D_r(1)$, il est possible d'avoir une meilleure estimation en limite fluide, qui s'obtient en conditionnant $D_r(1)$ sur le rang complet normalisé du premier partenaire acceptable [56]. On obtient alors

$$D_r(1) = \mathrm{e}E_1(1) \approx 0.596, \tag{4.17}$$

où E_1 est l'exponentielle intégrale.

Cette valeur est très bonne dans à peu près tous les cas pour les préférences acycliques aléatoires. Pour les préférences géométriques, il est en revanche nécessaire d'être dans de bonnes conditions, c'est-à-dire p petit et n grand [56].

4.4.5 Généralisation au b-couplage

Je propose maintenant d'étendre les résultats précédents au problème du b-couplage. b est toujours supposé constant sur les pairs pour plus de simplicité. Je ne propose ici que des résultats sur le rang complet, bien qu'il soit a priori possible de réutiliser les techniques vues précédemment pour le rang relatif ou les distances.

4.4.5.1 Équations sous l'hypothèse d'indépendance

Un pair peut maintenant avoir jusqu'à b partenaires. On note donc, pour $1 \leqslant c \leqslant b$, D_c la distribution du rang complet du $c^{ième}$ meilleur partenaire stable, et S_c la CCDF correspondante. Tout comme pour le cas $b = 1$, il est possible de donner les conditions nécessaires et suffisantes pour que $j = R_i^{-1}(K)$ soit le $c^{ième}$ meilleur partenaire stable de i :

- le couple $\{i, j\}$ doit être acceptable,
- le $(c-1)^{ième}$ meilleur partenaire de i (si $c > 1$) doit être strictement meilleur que j, tandis que le $c^{ième}$ (s'il existe) ne doit pas l'être,
- le $b^{ième}$ partenaire de j, s'il existe, ne doit pas être meilleur que i.

En étendant l'hypothèse d'indépendance 1, on obtient ainsi une formule générique pour le b-couplage :

$$D_{R_i,c}(K) = \begin{cases} pS_{R_i,1}(K)S_{R_j,b}(R_j(i)) \text{ si } c = 1, \text{ sinon} \\ p(S_{R_i,c}(K) - S_{R_i,c-1}(K))S_{R_j,b}(R_j(i)). \end{cases} \tag{4.18}$$

Pour être utilisable, cette formule doit être adaptée au type de préférences considéré.

Pour les préférences globales, si $D_c(i, j)$ désigne la probabilité que le $c^{ième}$ partenaire stable de i soit j [16], on obtient le système récursif suivant, qui se résout par une triple itération sur i, j et c [33] :

$$D_c(i, j) = \begin{cases} 0 \text{ si } i = j, \text{ sinon} \\ pS_1(i, j)S_b(j, i) \text{ si } c = 1, \\ p(S_c(i, j) - S_{c-1}(i, j))S_b(j, i) \text{ si } c > 1. \end{cases} \tag{4.19}$$

De même, pour les préférences acycliques aléatoires et géométriques, on obtient le système suivant à partir de l'approximation 2 (symétrie du rang complet) [56] :

$$D_{R,c}(K) = \begin{cases} pS_{R,1}(K)S_{R,b}(K) \text{ si } c = 1, \\ p(S_{R,c}(K) - S_{R,c-1}(K))S_{R,b}(K) \text{ si } c > 1. \end{cases} \tag{4.20}$$

En utilisant $S_{R,c}(1) = 1$ et $D_{R_c}(K) = S_{R,c}(K) - S_{R,c}(K + 1)$, le système (4.20) se résout facilement par une double itération sur K et c.

16. Je me permets de souligner au passage que D_c n'est plus symétrique, alors que c'était le cas pour $b = 1$.

Les simulations montrent que les deux systèmes coïncident très bien avec les distributions empiriques tant que n et p vérifient les conditions usuelles [56]. Le comportement est qualitativement très similaire à celui du couplage simple : décroissance exponentielle pour les préférences globales (mais à cause de la multiplicité des partenaires, des décalages apparaissent entre les pics de densité des D_c et la valeur du pair d'origine), distribution en aile lourde pour les préférences géométriques et acycliques aléatoires.

4.4.5.2 Limites fluides

Pour $b > 1$, des limites fluides existent aussi. Hélas, contrairement au couplage simple, je n'ai pas réussi à trouver de solutions explicites pour les décrire (et le temps passant, je doute de plus en plus que de telles solutions existent). Il est néanmoins possible de donner les EDP vérifiées par ces limites.

Pour les préférences globales, les limites fluides vérifient

$$\partial_y \mathcal{S}_c(\alpha,\beta) = \begin{cases} -d\mathcal{S}_1(\alpha,\beta)\mathcal{S}_b(\beta,\alpha) \text{ si } c=1, \text{ sinon} \\ -d\left(\mathcal{S}_c(\alpha,\beta) - \mathcal{S}_{c-1}(\alpha,\beta)\right)\mathcal{S}_b(\beta,\alpha), \end{cases} \tag{4.21}$$

avec les conditions initiales $\mathcal{S}_c(\alpha,0) = 1$.

De même, en préférences acycliques et géométriques, les limites $\mathcal{S}_{R,c}$ vérifient

$$\dot{\mathcal{S}}_{R,c} = \begin{cases} -d\mathcal{S}_{R,1}\mathcal{S}_{R,b} \text{ si } c=1, \\ -d(\mathcal{S}_{R,c} - \mathcal{S}_{R,c-1})\mathcal{S}_{R,b} \text{ si } c>1, \end{cases} \tag{4.22}$$

avec les conditions initiales $\mathcal{S}_{R,c}(0) = 1$.

Même si ces équations ne peuvent pas être résolues complètement, la limite fluide a quand même plusieurs intérêts.

Tout d'abord, l'existence même d'une limite permet de la calculer numériquement avec précision et d'utiliser le résultat pour de multiples valeurs de n et de d. Prenons par exemple le cas des préférences globales (le même raisonnement pouvant être fait pour les préférences acycliques et géométriques). On suppose b fixé. Soit $d_{\max}$ le degré moyen maximal des distributions que l'on veut évaluer, et N une taille fixée d'échantillonnage de la limite fluide (plus N est grand, plus l'évaluation sera précise). On peut alors poser $p = \frac{d_{\max}}{N-1}$ et calculer les $D_{N,p,c}$ (une fois pour toutes) en utilisant (4.19). Pour un degré $d \leqslant d_{\max}$, on pose $N' = \frac{d}{p}$ (N' n'est pas nécessairement entier). En s'inspirant de la normalisation (4.5), on obtient l'approximation suivante pour $|\alpha - \beta| \geqslant \frac{1}{N'}$:

$$\mathcal{D}_{d,c}(\alpha,\beta) \approx N' D_{N,p,c}(\lfloor N'\alpha \rfloor + 1, \lfloor N'\beta \rfloor + 1). \tag{4.23}$$

On peut alors en retour utiliser cette estimation de la limite fluide pour des distributions discrètes. Ainsi, pour tout entier n et pour tout $d \leqslant d_{\max}$, alors on a, pour $|i-j| \geqslant \frac{n}{N'}$ (ce qui revient à $i \neq j$ si $n \leqslant N'$)

$$D_{n,d,c}(i,j) \approx \frac{N'}{n} D_{N,p,c}(\lfloor \frac{N'}{n}(i-1) \rfloor + 1, \lfloor \frac{N'}{n}(j-1) \rfloor + 1). \tag{4.24}$$

Un autre intérêt des limites fluides est que les EDP qu'elles vérifient nous renseignent sur leur comportement. On peut par exemple s'en servir pour montrer que $\mathcal{S}_{R,1} \leqslant \mathcal{S}_R \leqslant \mathcal{S}_{R,b}$ (et l'équivalent en préférences globales) et comprendre ainsi pourquoi le comportement en b-couplage reste similaire à celui du couplage simple.

4.4.6 Quelques applications

Ayant mis beaucoup d'efforts dans la simple étude des distributions, je dois reconnaître n'avoir pas encore consacré énormément de temps aux propriétés évoluées des configurations stables, c'est-à-dire à celles susceptibles d'aider à comprendre les systèmes existants et à en développer de nouveaux. En voici néanmoins deux : la stratification des préférences globales et la « petit-mondisation » des préférences géométriques.

4.4.6.1 Stratification

Comme nous venons de le voir, en préférences globales, les partenaires stables d'un pair i donné ont, en moyenne, le même rang que i. C'est la stratification, qui garantit une certaine équité dans la configuration stable [33] : en terme de rang, ce que donne un pair devrait être à peu près égal à ce qu'il reçoit. Il faut aussi se rappeler que les $\mathcal{D}_c(\alpha, .)$ ont une décroissance exponentielle avec déviation en $\frac{1}{d}$, d étant le degré moyen du graphe d'acceptabilité. On s'aperçoit alors qu'il faut résoudre le compromis suivant :

- si d est trop petit la déviation est élevée. En particulier, si les entrées de la matrice des valeurs (par exemple les bandes passantes d'un système à la *BitTorrent*) suivent une distribution non-uniforme, il peut y avoir une très grande différence entre l'espérance du gain et ce que l'on donne. Ce problème a été mis en évidence dans [33] pour expliquer des failles potentielles de la technique de Tit-for-Tat employée par BitTorrent ;
- un grand d permet à l'opposé de renforcer l'équité. Mais en pratique, augmenter la taille du graphe d'acceptabilité a un coût pour les pairs : temps de convergence, place mémoire, maintenance du graphe... Un grand d raccourcit aussi la déviation (jusqu'au Dirac dans la limite fluide), et donc augmente le diamètre de la configuration stable, ce qui peut être problématique si l'on veut propager de l'information à travers les arêtes stables.

Pour le quota b, qui représente le degré maximal dans la configuration stable, un compromis similaire existe : un grand b peut améliorer l'équité et diminuer le diamètre, mais va être coûteux en ressources.

Cela suggère que pour la plupart des systèmes à préférences globales (i.e. basés sur le partage de la bande passante, de la capacité de stockage ou de calcul, de l'uptime...), il devrait exister un couple (d, b) (ou plus généralement un couplage entre un graphe d'acceptabilité et un vecteur de quotas) optimal pour la configuration stable, dont la valeur exacte dépendrait de l'importance donnée à des paramètres comme le diamètre, l'équité, le temps de convergence ou le coût de maintenance.

4.4.6.2 Petit-mondisation

Un petit monde est un graphe creux (degré moyen en $O(\log(n))$ voire en $O(1)$) avec un plus court chemin moyen (*average shortest path length* ou ASPL) en $O(\log(n))$ et un coefficient de clustering élevé (il existe beaucoup plus de cycles courts que pour un graphe aléatoire de même taille). Par exemple, Kleinberg a montré il y a quelques années qu'une grille de dimension n pouvait être transformée en petit-monde à condition d'ajouter des arêtes longues suivant une distribution en $\Omega(\frac{1}{x^n})$ [47].

Regardons maintenant ce qui se passe pour une configuration stable, avec un b en $O(\log(n))$ pour que la configuration soit « creuse ». S'il s'agit de préférences globales, le clustering est bien là comme conséquence de la stratification, mais le diamètre a tendance à croître linéairement [32]. De même, pour les préférences acycliques aléatoires, le petit diamètre est vérifié mais il n'y a pas de grand clustering (la configuration stable se comporte comme un b-graphe aléatoire incomplet). Par contre, pour les préférences géométriques, la distribution en aile lourde permet d'avoir les deux propriétés : d'un côté, la plupart des partenaires stables ont un petit rang complet ; ils sont donc « géographiquement » proches ce qui donne le clustering ; de l'autre, il existe des liens longs qui rendent le diamètre petit. Les préférences géométriques sont donc propices à la génération de configurations en petit-monde, et c'est effectivement ce qui se produit [32]. Cela ouvre un certain nombre de perspectives sur l'utilisation de la configuration stable, à commencer par un générateur de petits-mondes facile à utiliser.

Ce qui est étonnant dans cette petit-mondisation en préférences géométriques, c'est qu'elle est uniquement créée par la manière dont les pairs se classent les uns les autres : les distances réelles sont utilisées pour construire ces classements, mais les valeurs en elles-mêmes ne jouent aucun rôle direct dans le système à préférences en général et dans sa configuration stable en particulier. Ceci amène à penser que des caractéristiques topologiques d'un système peuvent donc être contenues dans un ensemble de préférences.

Comme exemple de caractéristiques, je me suis amusé à calculer numériquement les paramètres de petit-monde des préférences sur les tores pour quelques dimensions, le résultat étant que le diamètre et le clustering tendent à baisser quand la dimension augmente [56]. J'ai ensuite regardé ce que donnaient les latences Meridian, et les paramètres obtenus se sont révélés proches de ceux du tore de dimension 3. J'aime particulièrement ce résultat inattendu, car il semble suggérer qu'il existe une dimension d'Internet *au sens des préférences*, qui vaut à peu près 3. Je peux ainsi rajouter ma pierre au mur des efforts fournis un peu partout pour estimer une dimension d'Internet (voir par exemple [8]).

4.5 Conclusion

Après une rapide histoire des mariages stables, j'ai présenté les bases nécessaires au maniement des préférences acycliques : le vocabulaire spécifique (parfois imagé), le formalisme, le grand théorème d'*existence, unicité, convergence* et une description des principales classes de préférences acycliques, comme les préférences globales ou géométriques.

J'ai ensuite détaillé la propriété d'auto-stabilisation, et plus précisément le temps nécessaire pour une convergence complète. J'ai présenté des techniques qui permettent de borner ou d'estimer ce temps pour certains paramètres particuliers, et j'ai proposé l'utilisation de simulations pour mettre en évidence le rôle joué par chaque paramètre. On retiendra que dans le pire des cas, avec des pairs asynchrones, la convergence peut être très longue, alors qu'en pratique, elle est assez rapide (préférences globales), voire très rapide (la plupart des autres préférences).

Enfin, j'ai décrit la distribution des partenaires stables pour quelques classes de préférences, en proposant une analyse de cette distribution sous des hypothèses d'indépendance, puis en validant et étendant ces résultats numériquement. Pour les préférences globales, j'ai ainsi mis en évidence l'effet de *stratification*. De manière imagée, dans un monde où il n'y aurait qu'un unique critère de beauté, les beaux se retrouvent avec les beaux, et les moches n'ont guère d'autre choix que d'être avec les moches [17].

Pour les préférences géométriques, c'est une petit-mondisation qui se produit, due au fait que la distribution des partenaires est en aile lourde, ce qui assure à la fois la localité et les liens longs. L'image est celle d'un monde régi par l'homophilie (*qui se ressemble s'assemble*) : la plupart du temps, la réciprocité permet des collaborations entre proches, mais quand un pair est rejeté par ses plus proches voisins, il doit chercher de plus en plus loin, chez des partenaires potentiels qui l'apprécient de moins en moins au fur et à mesure qu'il doit s'éloigner de sa propre position.

Au final, j'espère avoir réussi à faire partager ma conception des réseaux à préférences acycliques : un modèle possible pour les réseaux pair-à-pair, un ensemble varié de petits problèmes mathématiques, mais aussi une informatique récréative. Je suis conscient que ce chapitre pose plus de questions qu'il n'apporte de réponses : modélisation de l'initiative aléatoire, de l'agitation, étude de nouvelles préférences (pourquoi pas cycliques), graphes et quotas arbitraires, voire variables, routage dans les petits-mondes géométriques... Je n'aurais assurément ni le temps, ni les facultés de répondre à tous ces problèmes, et en conséquent, si ce chapitre donne l'envie à au moins un lecteur de s'intéresser aux réseaux à préférences acycliques, je pense qu'il aura parfaitement rempli son rôle.

Pour ma part, je pense dans le futur essayer d'appliquer les réseaux à préférences à des situations pratiques, et les relier aux travaux que j'ai effectué sur les problématique de la distribution. J'estime que les préférences acycliques commencent à avoir la maturité nécessaire pour apporter un éclairage nouveau à ces problèmes, et j'aimerais le vérifier, comme j'en parle dans le prochain et dernier chapitre.

17. Dans la même veine, je conseille l'article *Beauty and distance in the stable marriage problem*, qui pimente l'algorithme du bal en étudiant l'effet de l'introduction d'un critère de beauté [15].

Chapitre 5

Synthèse et perspectives

J'ai présenté au cours de ces quelques pages le point de vue qui est le mien sur ce nouveau domaine de recherche qu'est le pair-à-pair. Partant d'une définition originale qui a l'avantage de mettre en avant pourquoi le pair-à-pair est un domaine de recherche, j'ai proposé une classification qui permet de décrire, dans les grandes lignes, les différents thèmes qu'il est possible d'aborder, suivant que l'on considère le problème de la localisation ou de la distribution ; qu'on l'aborde à l'aide d'une structure explicite ou implicite ; qu'on place la décision à l'émission, à la réception ou des deux côtés ; qu'on ait une approche initiale théorique ou empirique.

J'ai ensuite placé mes propres contributions au sein de ces thématiques : alors qu'une partie de mes travaux s'inscrit dans le cadre d'une meilleure compréhension des problèmes liés à la diffusion, tout en ouvrant la voie à de futures et prometteuses applications, une autre partie est consacrée à créer un nouveau modèle permettant d'apporter un autre éclairage sur les dynamiques des systèmes auto-structurés en général, et du pair-à-pair en particulier.

Je n'ai pas du tout parlé de mes travaux faits en continuité de ma thématique de thèse, le PageRank [14, 59, 24, 25, 37, 38], car bien que traitant d'un sujet proche, les grands graphes, ils ne rentrent pas à proprement parler dans le cadre du pair-à-pair, et ce n'est pas mon principal sujet d'étude à l'heure actuelle. Néanmoins, il n'est pas impossible qu'un nouvel hasard me fasse un jour découvrir des nouvelles thématiques liées à ce sujet et relance complètement mon intérêt.

Pour revenir au sujet du pair-à-pair, j'aimerais insister une dernière fois sur le fait que les différentes thématiques ne sont pas figées ni cloisonnées, et que si cela rend la classification des recherches plus analogique que dichotomique, je pense que cela est *in fine* bénéfique. Par exemple, certains graphes, comme les graphes de de Bruijn, ont été introduits en pair-à-pair afin de réaliser des DHTs [31]. Plus tard, ces mêmes graphes se sont échappés du problème de la localisation et ont été utilisés pour proposer des structures de diffusion [35]. De même, les techniques épidémiques, qui sont par nature des techniques de diffusion, peuvent avoir une utilité en localisation [44]. Au fil des ans, cette porosité permet ainsi de faire progresser le front de la recherche, et j'aime à croire que cela est loin d'être terminé.

Car tous ces échanges correspondent parfaitement à la philosophie que j'ai adopté en tant que chercheur, et mon intuition me dit qu'en restant à l'écoute des innovations et des mutations qui se produisent dans tous ces thèmes, j'ai des chances de découvrir de nouvelles pistes qui éveilleront ma curiosité. Il y en a certaines sur lesquelles j'ai une certaine idée, et j'aimerais conclure ce mémoire en exposant ces quelques axes sur lesquels je pense travailler à moyen terme, sachant qu'il est toujours possible que demain, une nouvelle rencontre provoque une nouvelle orientation de mes recherches.

Réseaux à préférences

Comme le lecteur s'en sera rendu compte, le sujet des réseaux à préférences me tient à cœur, et je pense qu'il devrait encore occuper une partie de mes travaux pour quelques temps encore.

Le chemin qu'il reste à parcourir est en effet au moins aussi long, et je le pense aussi passionnant, que celui qui a déjà été fait : caractérisation plus fine des propriétés auto-stabilisantes, modélisation des graphes et quotas dynamiques, étude des propriétés de routage, retour aux préférences cycliques... J'espère, en continuant d'en faire la promotion au sein de la communauté scientifique, réussir à évangéliser des chercheurs qui à leur tour feront progresser la théorie tout en développant des applications concrètes.

Sur le front de ces applications concrètes, j'estime qu'il doit être possible d'utiliser le modèle afin de concevoir des algorithmes de diffusion épidémique encore plus performants. En effet, comme je l'ai montré, les préférences globales permettent une sélection par la bande passante, et des travaux récents semble désigner la sélection orientée par la bande passante comme prometteuse pour réaliser une diffusion épidémique au sein de pairs de capacités hétérogènes. De même, les résultats obtenus sur les préférences de latences permettent d'espérer une minimisation de l'impact réseau de la diffusion sans affecter les propriétés du graphe de diffusion. En combinant de la bonne manière ces deux types de préférences et en appliquant le résultat à la diffusion épidémique, l'algorithme final a de bonnes chances d'être très efficace.

Au-delà du pair-à-pair, j'espère aussi découvrir de nouvelles choses en regardant si les réseaux à préférences peuvent avoir un lien avec les réseaux de capteurs, sur lesquels je compte me pencher. Ce glissement de thématique sera je pense bénéfique pour les réseaux à préférences, et qui sait peut-être aussi pour les réseaux de capteurs.

Prototypes

C'est un fait, je ne suis pas un développeur. Malgré tout, je trouve extrêmement frustrant d'avoir travaillé sur des concepts et des algorithmes sans pouvoir vérifier s'ils tiennent la route en pratique. Si je veux être en accord avec mon positionnement en tant que chercheur, je pense donc que je me dois d'essayer de pousser les travaux qui peuvent l'être le plus loin possible sur le chemin de l'application, pourquoi pas jusqu'au déploiement. C'est bien sûr une tâche que je suis incapable de réaliser seul, et j'espère surtout arriver à convaincre des spécialistes du domaine de réaliser le développement.

À l'heure actuelle, il existe deux prototypes que j'apprécierais d'avoir à ma disposition afin de pouvoir tester *grandeur nature* et améliorer les algorithmes sur lesquels j'ai travaillé jusqu'à présent. Il n'est pas très difficile de deviner lesquels : un prototype de diffusion épidémique, et un prototype de vidéo-à-la-demande.

Futur du pair-à-pair

Pour conclure sur un axe plus général, j'aimerais partager mon opinion sur le devenir du pair-à-pair. Car si le pair-à-pair doit évoluer, la recherche devra s'adapter elle aussi, ceci incluant mes propres travaux.

Revenons un instant aux origines de l'*âge d'or* du pair-à-pair, telles que je les ai décrites dans l'introduction : d'un point de vue social, l'explosion du pair-à-pair a été provoqué par le besoin de fournir un service pour lequel la demande existe (conséquence des progrès techniques et de la bulle Internet), mais pas l'offre (éclatement de la bulle). Mais aujourd'hui, les services sont à nouveau offerts : espaces de stockage, partage et commerce de fichiers ou de contenus multimedia, télévision et magnétoscope sur Internet... Grâce à une ergonomie supérieure, ces services tendent à reprendre du terrain sur le pair-à-pair en tant que phénomène de société. Le véritable enjeu de cette bataille est la localisation des ressources, chez l'usager (pair-à-pair) ou sur des serveurs (Google, pour simplifier). L'issue en est encore incertaine, avec des victoires et des défaites de part et d'autre. Ce qui est sûr, c'est que nous sommes dans une situation transitoire et que les usages actuels, en particulier les usages pair-à-pair, sont condamnés à évoluer ou à dépérir.

Mais *in fine*, il y a aura toujours d'un côté des services à fournir, et de l'autre des ressources, fussent-elles sur un PC, un téléphone, un magnétoscope virtuel, un adjoint de poste de télévision,

un proxy ou un nuage de serveurs. Avec la multiplication des supports, côté usager ou côté fournisseur, il y a donc une question qui, à mon avis, va rester d'actualité quoi qu'il arrive : *qui récupère quoi de la part de qui ?* Et si l'on me demande mon avis sur l'avenir du pair-à-pair (et donc accessoirement sur celui des recherches sur le sujet), je mise sur sa prévisible métempsychose dans les technologies à venir, quelles qu'elles soient. Autrement dit, s'il est possible que le pair-à-pair, dans sa compréhension bittorrentesque, finisse par disparaître dans les années à venir, la recherche en pair-à-pair, qui consiste à répondre à la question *qui récupère quoi de la part de qui ?* quand elle n'est pas triviale, a encore de beaux jours devant elle. L'appellation changera peut-être après absorption par un nouveau domaine émergeant (Cloud Computing ? Virtualisation ?) mais je crois que l'esprit, les thématiques et les méthodes du domaine survivront à la transition. Cependant, cela ne veut pas dire non plus que le domaine ne doit pas anticiper sa propre évolution. Certaines de ces directions sont d'ailleurs déjà visibles : apparition de nouveaux services qui posent de nouveaux défis, par exemple le *start-over* qui combine diffusion en léger différé et à la demande ; hétérogénéité croissante des ressources disponibles, avec l'apparition de solutions hybrides et la multiplication des supports ; déplacement des goulots d'étranglement lié aux changements dans le réseau et les usages.

Bibliographie

[1] TVants, http ://tvants.en.softonic.com/.

[2] Babelgum, http ://www.babelgum.com/.

[3] Joost, http ://www.joost.com/.

[4] Kontiki, http ://www.kontiki.com/.

[5] Sopcast, http ://www.sopcast.com/.

[6] UUsee inc., http ://www.uusee.com/.

[7] David J. Abraham, Ariel Levavi, David Manlove, and Gregg O'Malley. The stable roommates problem with globally-ranked pairs. In *WINE*, pages 431–444, 2007.

[8] B. Abrahao and R. Kleinberg. On the internet delay space dimensionality. In *Proceedings of the 2008 Internet Measurement*, 2008.

[9] Richard Bellman. The stability of solutions of linear differential equations. *Duke Math. J.*, 10 :643–647, 1943.

[10] Farid Benbadis, Nidhi Hegde, Fabien Mathieu, and Diego Perino. Playing with the bandwidth conservation law. In *8th International Conference on Peer-to-Peer Computing*, pages 140–149, 2008.

[11] Thomas Bonald, Laurent Massoulié, Fabien Mathieu, Diego Perino, and Andrew Twigg. Epidemic live streaming : optimal performance trade-offs. In *Proceedings of the International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, 2008.

[12] Yacine Boufkhad, Fabien Mathieu, Fabien de Montgolfier, Diego Perino, and Laurent Viennot. Achievable catalog size in peer-to-peer video-on-demand systems. In *Proceedings of the Seventh International Workshop on Peer-to-Peer Systems (IPTPS)*, 2008.

[13] Yacine Boufkhad, Fabien Mathieu, Fabien de Montgolfier, Diego Perino, and Laurent Viennot. An upload bandwidth threshold for peer-to-peer video-on-demand scalability. In *Proceedings of the 23rd IEEE International Parallel & Distributed Processing Symposium (IPDPS)*, 2009.

[14] Mohamed Bouklit and Fabien Mathieu. Backrank : an alternative for pagerank ? In *Proceedings of WWW '05 : Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 1122–1123, New York, NY, USA, 2005. ACM.

[15] G. Caldarelli and A. Capocci. Beauty and distance in the stable marriage problem. *Physica A Statistical Mechanics and its Applications*, 300 :325–331, November 2001.

[16] M. Castro, P. Druschel, A. Kermarrec, A. Nandi, A. Rowstron, and A. Singh. Splitstream : High-bandwidth multicast in cooperative environments. In *Proc. of the 19th ACM Symp. on Operating Systems Principles (SOSP)*, 2003.

[17] K. Cechlárová and T. Fleiner. On a generalization of the stable roommates problem. *ACM Trans. Algorithms*, 1(1) :143–156, 2005.

[18] Sarvadaman Chowla. The asymptotic behavior of solutions of difference equations. In *Proceedings of the International Congress of Mathematicians, Vol. I, 377, Amer. Math. Soc.*, 1950.

[19] Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong. Freenet : A distributed anonymous information storage and retrieval system. *Lecture Notes in Computer Science*, 2009 :46–66, 2001.

[20] B. Cohen. Incentives build robustness in bittorrent. In *P2PECON*, 2003.

[21] R. Piegay A.M. Kermarrec E. Anceaume F. Le Fessant D. Frey, J. Royan. Solipsis : A decentralized architecture for virtual environments. In *International Workshop on Massively Multiuser Virtual Environments*. IEEE, Mar 2008.

[22] E. Diamantoudi, E. Miyagawa, and L. Xue. Random paths to stability in the roommate problem. *Games and Economic Behavior*, 48(1) :18–28, July 2004.

[23] Edsger W. Dijkstra. Self-stabilizing systems in spite of distributed control. *Commun. ACM*, 17(11) :643–644, 1974.

[24] Saint-Jean Djungu, Pierre Manneback, and Fabien Mathieu. étude comparative des méthodes de calcul de pagerank. In *Proceedings of the 8ème Colloque Africain sur la Recherche en Informatique*, 2006.

[25] Saint-Jean Djungu, Pierre Manneback, and Fabien Mathieu. Speedsiterank : Algorithme parallèle pour un pagerank distribué en sites. In *Proceedings of the 9ème Colloque Africain sur la Recherche en Informatique*, 2008.

[26] Dominique Dumont. Mariages stables. *Séminaire Lotharingien de Combinatoire*, 23, 1999.

[27] http ://www.edonkey2000.com/index.html.

[28] http ://www.emule-project.net/.

[29] Fabrice Le Fessant, Simon Patarin, and Le Chesnay Cedex. Mldonkey, a multi-network peer-to-peer file-sharing program. Technical Report RR–4797, INRIA, 2003.

[30] T. Fleiner. The stable b-matching polytope. *Mathematical Social Science*, 46(2) :149–158, 2003.

[31] Pierre Fraigniaud and Philippe Gauron. D2b : a de bruijn based content-addressable network. *Theor. Comput. Sci.*, 355(1) :65–79, 2006.

[32] Anh-Tuan Gai, Dmitry Lebedev, Fabien Mathieu, Fabien de Montgolfier, Julien Reynier, and Laurent Viennot. Acyclic preference systems in P2P networks. In *Euro-Par*, pages 825–834, 2007.

[33] Anh-Tuan Gai, Fabien Mathieu, Fabien de Montgolfier, and Julien Reynier. Stratification in p2p networks : Application to bittorrent. In *ICDCS*, pages 30–39. IEEE Computer Society, 2007.

[34] A.T. Gai and L. Viennot. Broose : a practical distributed hashtable based on the de-brujin topology. In *Proc. of the 4th IEEE Int. Conf. on Peer-to-Peer Computing (P2P)*, 2004.

[35] A.T. Gai and L. Viennot. Incentive, resilience and load balancing in multicasting through clustered de bruijn overlay network. In *14th IEEE International Conference On Networks (ICON)*, september 2006.

[36] D. Gale and L.S. Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 69 :9–15, 1962.

[37] Bruno Gaume and Fabien Mathieu. From random graph to small world by wandering. Technical Report RR-6489, INRIA, 2008.

[38] Bruno Gaume and Fabien Mathieu. Pagerank-induced topology for real-world networks. *Submitted to Journal of Complex Systems*, 2009.

[39] X. Hei, C. Liang, J. Liang, Y. Liu, and K. W. Ross. Insights into PPLive : A measurement study of a large-scale P2P IPTV system. In *Proc. of IPTV Workshop, International World Wide Web Conference*, 2006.

[40] R. Irving, P. Leather, and D. Gusfield. An efficient algorithm for the "optimal" stable marriage. *J. ACM*, 34(3) :532–543, 1987.

[41] Robert W. Irving, David Manlove, and Sandy Scott. The hospitals/residents problem with ties. In *SWAT '00 : Proceedings of the 7th Scandinavian Workshop on Algorithm Theory*, pages 259–271, London, UK, 2000. Springer-Verlag.

[42] Robert W. Irving and David F. Manlove. The stable roommates problem with ties. *J. Algorithms*, 43(1) :85–105, 2002.

[43] K. Iwama, S. Miyazaki, D. Manlove, and Y. Morita. Stable marriage with incomplete lists and ties. In *ICALP*, pages 443–452, 1999.

[44] Holger Kampffmeyer, Mirco Musolesi, and Cecilia Mascolo. Epitella : Improving the gnutella search algorithm through epidemic spreading models for complex networks. In *Proceedings of the IEEE Workshop on Dependable Support in Self-Organising Networks (DASSON'07). Colocated with 37th Annual IEEE/IFIP Conference on Dependable Systems and Networks*, Edinburgh, United Kingdom, June 2007. IEEE Press.

[45] Y. Kawahara, T. Aoyama, and H. Morikawa. A peer-to-peer message exchange scheme for large-scale networked virtual environments. *Telecommunication Systems*, 25(3), 2004.

[46] Joaquin Keller and Gwendal Simon. Solipsis : a massively multi-participant virtual world. In *Intern. Conf. on Parallel and Distributed Techniques and Applications*, 2003.

[47] Jon Kleinberg. The small-world phenomenon : an algorithm perspective. In *STOC '00 : Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 163–170, New York, NY, USA, 2000. ACM.

[48] F. Le Fessant, S. Handurukande, A.-M. Kermarrec, and L. Massoulié. Clustering in peer-to-peer file sharing workloads. In *IPTPS*, 2004.

[49] Dmitry Lebedev, Fabien Mathieu, Laurent Viennot, Ahn-Tuan Gai, Julien Reynier, and Fabien de Montgolfier. On using matching theory to understand P2P network design. In *International Network Optimization Conference (INOC)*, 2007.

[50] Y.-J. Lin, K. Guo, and S. Paul. Sync-ms : synchronized messaging service for real-time multi-player distributed games. In *Proc. of the 10th IEEE International Conference on Network Protocols*, 2002.

[51] D. Manlove. The structure of stable marriage with indifference. *Dis. Appl. Math.*, 122 :167–181, 2002.

[52] David F. Manlove, Robert W. Irving, Kazuo Iwama, Shuichi Miyazaki, and Yasufumi Morita. Hard variants of stable marriage. *Theor. Comput. Sci.*, 276(1-2) :261–279, 2002.

[53] Fabien Mathieu. Upper bounds for stabilization in acyclic preference-based systems. In *SSS*, pages 372–382, 2007.

[54] Fabien Mathieu. Self-stabilization in preference-based systems. *Peer-to-Peer Networking and Applications*, 1(2) :104–121, sept 2008.

[55] Fabien Mathieu. Acyclic preference-based systems. In X. Shen, H. Yu, J. Buford, and M. Akon, editors, *Hanbook of Peer-to-Peer Networking*. Springer, 2009.

[56] Fabien Mathieu, Gheorghe Postelnicu, and Julien Reynier. The stable configuration in acyclic preference-based systems. Technical Report RR-6628, INRIA, 2008.

[57] Fabien Mathieu, Gheorghe Postelnicu, and Julien Reynier. The stable configuration in acyclic preference-based systems. In *28th IEEE International Conference on Computer Communications (INFOCOM)*, 2009.

[58] Fabien Mathieu and Julien Reynier. Missing piece issue and upload strategies in flashcrowds and p2p-assisted filesharing. In *AICT-ICIW '06 : Proceedings of the Advanced Int'l Conference on Telecommunications and Int'l Conference on Internet and Web Applications and Services*, page 112, Washington, DC, USA, 2006. IEEE Computer Society.

[59] Fabien Mathieu and Laurent Viennot. Local aspects of the global ranking of web pages. In *Proceedings of the 6th International Workshop on Innovative Internet Community Systems*, 2006.

[60] P. Maymounkov and D. Mazieres. Kademlia : A peer-to-peer information system based on the xor metric. In *1st International Workshop on Peer-to-Peer Systems (IPTPS)*, 2002.

[61] Meridian Project. http ://www.cs.cornell.edu/People/egs/meridian/.

[62] D. Qiu and R. Srikant. Modeling and performance analysis of BitTorrent-like Peer-to-Peer networks. In *INFOCOM*, 1998.

[63] D. Qiu and R. Srikant. Modeling and performance analysis of bittorrent-like peer-to-peer networks, 2004.

[64] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Schenker. A scalable content-addressable network. In *SIGCOMM '01 : Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 161–172, New York, NY, USA, 2001. ACM.

[65] Eytan Ronn. *On the complexity of stable matchings with and without ties.* PhD thesis, Yale University, 1986.

[66] Alvin E Roth. The evolution of the labor market for medical interns and residents : A case study in game theory. *Journal of Political Economy*, 92(6) :991–1016, 1984.

[67] Alvin E. Roth, Tayfun Sonmez, and M. Utku Unver. Pairwise kidney exchange. *Journal of Economic Theory*, 125(2) :151–188, December 2005. available at http ://ideas.repec.org/a/eee/jetheo/v125y2005i2p151-188.html.

[68] A. Rowstron and P. Druschel. Pastry : Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. *Lecture Notes in Computer Science*, 2218 :329–350, 2001.

[69] Fabrice Schuler. Étude et utilisation des technologies des p2p. `http://schuler.developpez.com/articles/p2p/`, 2005.

[70] X. Shen, H. Yu, J. Buford, and M. Akon, editors. *Handbook of Peer-to-Peer Networking.* Springer, 2009.

[71] K. Sripanidkulchai, B. Maggs, and H. Zhang. Efficient content location using interest-based locality in peer-to-peer systems. In *INFOCOM*, 2003.

[72] Ralf Steinmetz and Klaus Wehrle, editors. *Peer-to-Peer Systems and Applications*, volume 3485 of *Lecture Notes in Computer Science.* Springer, 2005.

[73] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord : A scalable peer-to-peer lookup service for internet applications. In *SIGCOMM '01 : Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 149–160, New York, NY, USA, 2001. ACM.

[74] Kyoungwon Suh, Christophe Diot, Jim Kurose, Laurent Massoulie, Christoph Neumann, Donald F. Towsley, and Matteo Varvello. Push-to-peer video-on-demand system : Design and evaluation. *IEEE Journal on Selected Areas in Communications*, 25(9) :1706–1716, 2007.

[75] Yu-Wei Sung, Michael Bishop, and Sanjay Rao. Enabling contribution awareness in an overlay broadcasting system. *SIGCOMM Comput. Commun. Rev.*, 36(4) :411–422, 2006.

[76] L. Viennot, R. Cardin, A. Leveillé, R. Chabassol, A. Gomart, A. Moussay, O. Lausecker, and B. Hidoine. Il était une fois le peer-to-peer. Vulgarization movie on `http://interstices.info/display.jsp?id=c_8622`, 2004. Inria-Multimédia.

[77] Wikibook. The world of peer-to-peer (p2p). `http://en.wikibooks.org/wiki/The_World_of_Peer-to-Peer_(P2P)`, 2008.

[78] X. Zhang, J. Liu, B. Li, and T. Yum. Coolstreaming/donet : A data-driven overlay network for Peer-to-Peer live media streaming. In *INFOCOM*, 2005.

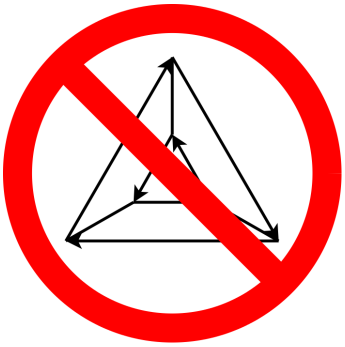
[79] Ben Y. Zhao, Ben Y. Zhao, John Kubiatowicz, John Kubiatowicz, Anthony D. Joseph, and Anthony D. Joseph. Tapestry : An infrastructure for fault-tolerant wide-area location and routing. Technical Report UCB/CSD-01-1141, U. C. Berkeley, 2001.

[80] Randall Zisk. There's No 'I' in Team. *Grey's Anatomy*, Season 5(Episode 4), October 2008.

Résumé

D'un millénaire à l'autre, le pair-à-pair a émergé comme un nouveau paradigme informatique. Plus précisément, des nouveaux enjeux sociaux et économiques, ayant trait en particulier à la distribution de contenus, sont venus raviver d'anciennes problématiques liées aux systèmes décentralisés, leur donnant de nouvelles justifications et de nouveaux éclairages.

Dans ce mémoire, je propose tout d'abord de donner des bases pour comprendre et aborder les problématiques pair-à-pair. Après un bref survol des travaux auxquels je me suis intéressé dans le but d'améliorer la distribution de contenu, je me tourne vers un sujet plus théorique : les réseaux à préférences acycliques, lesquels sont un moyen élégant pour modéliser des systèmes pair-à-pair non-structurés ou hybrides. Issus de la théorie des mariages stables, leur principale caractéristique est une capacité auto-stabilisante. Deux questions fondamentales se posent alors, auxquelles je donne un début de réponse : à quelle vitesse un réseau à préférences acycliques se stabilise-t-il, et vers quel état converge-t-il ?



Abstract

Peer-to-peer (P2P) recently emerged as a new paradigm in computer science. Due to major economic and social stakes, mostly related to content distribution, P2P has brought back to the forefront many existing research fields related to distributed systems, providing new incentives and goals.

In this work, we give some keys to the understanding of the research fields related to P2P systems. After a brief survey of our work on content distribution, we consider a more theoretical subject : acyclic preference-based systems, which recently appeared as an elegant way to model many P2P unstructured or hybrid systems. The strength of these models is a self-stabilizing property that allows us to provide analytical results in addition to empirical validation.