



**HAL**  
open science

# Impact du comportement des utilisateurs dans les réseaux pair-à pair, modélisation et simulation multi-agents

Julien Siebert

► **To cite this version:**

Julien Siebert. Impact du comportement des utilisateurs dans les réseaux pair-à pair, modélisation et simulation multi-agents. [Travaux universitaires] 2007, pp.41. inria-00172068

**HAL Id: inria-00172068**

**<https://inria.hal.science/inria-00172068>**

Submitted on 14 Sep 2007

**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.

# Impact du comportement des utilisateurs dans les réseaux pair à pair, modélisation et simulation multi-agents

## MÉMOIRE

soutenu le 27 juin 2007

pour l'obtention du

Master de l'Université Henri Poincaré – Nancy I

(Master Informatique - Spécialité : Perception Raisonement et Interactions  
Multimodales)

par

Julien Siebert

### Composition du jury

Noëlle Carbonell	Professeur
Didier Galmiche	Professeur
Dominique Méry	Professeur
Claude Godart	Professeur
Guy Perrier	Professeur

<i>Encadrants :</i>	Laurent Ciarletta	Maître de conférences
	Vincent Chevrier	Maître de conférences

# Table des matières

<b>Introduction</b>	<b>1</b>
<b>I Contexte de l'étude</b>	<b>2</b>
<b>1 Les réseaux Peer-to-Peer (P2P) d'échange de fichiers</b>	<b>3</b>
1.1 Présentation . . . . .	3
1.2 Propriétés . . . . .	3
1.3 État de l'art . . . . .	4
1.4 Kademia . . . . .	6
1.5 Impact du comportement des utilisateurs . . . . .	8
1.6 Modélisation et simulation des systèmes P2P . . . . .	9
1.7 Synthèse . . . . .	10
<b>2 Un mot sur l'approche par Système Multi-Agents (SMA)</b>	<b>11</b>
2.1 Notion d'agent . . . . .	11
2.2 SMA et applications . . . . .	12
2.3 Agents réactifs et cognitifs . . . . .	12
2.4 La simulation multi-agents . . . . .	13
2.5 Synthèse . . . . .	13
<b>3 Présentation des objectifs</b>	<b>14</b>
3.1 Synthèse des problématiques . . . . .	14
3.2 Objectifs . . . . .	14
<b>II De la modélisation à la simulation des réseaux P2P via une approche multi-agents</b>	<b>15</b>
<b>4 P2P et SMA : quels besoins en terme de simulation ?</b>	<b>16</b>
4.1 Analogie . . . . .	16
4.2 Une approche multi-modèles . . . . .	16
4.3 Modélisation de l'agent . . . . .	17
4.4 Modélisation de l'environnement . . . . .	17

4.5 Synthèse . . . . .	18
<b>5 Choix d'un outil de simulation</b>	<b>19</b>
5.1 Critères utilisés . . . . .	19
5.2 Les Simulateurs pair-à-pair existants . . . . .	20
5.3 Peerfactsim (PFS) . . . . .	20
5.4 Synthèse . . . . .	21
<b>6 Présentation du cas d'étude</b>	<b>22</b>
6.1 Problématique . . . . .	22
6.2 Définition des besoins . . . . .	22
<b>III Contributions</b>	<b>23</b>
<b>7 Modélisation du comportement de l'utilisateur</b>	<b>24</b>
7.1 Un modèle d'agent réactif . . . . .	24
7.2 Synthèse . . . . .	25
<b>8 Modification effectuées au sein du simulateur</b>	<b>26</b>
8.1 Du simulateur P2P au laboratoire virtuel . . . . .	26
8.2 Mesures . . . . .	28
8.3 Synthèse . . . . .	28
<b>9 Expérimentations</b>	<b>29</b>
9.1 Protocole expérimental . . . . .	29
9.2 Premières expérimentations . . . . .	29
9.3 Premiers résultats . . . . .	30
9.4 Passage à l'échelle de l'outil . . . . .	31
<b>Conclusion et perspectives</b>	<b>32</b>
1 Bilan . . . . .	32
2 Limites . . . . .	32
3 Perspectives . . . . .	33
<b>Table des figures</b>	<b>34</b>
<b>Liste des abréviations utilisées</b>	<b>35</b>
<b>Bibliographie</b>	<b>36</b>

# Remerciements

Je remercie le Loria en la personne de son directeur, Karl Tombre pour m'avoir accueilli à l'occasion de ce stage.

Je tiens aussi à remercier l'ensemble des personnes des équipes MADYNES et MAIA et plus particulièrement les chefs d'équipes respectifs : Olivier Festor et François Charpillat.

Je remercie aussi mes encadrants Laurent Ciarletta et Vincent Chevrier pour leurs précieux conseils et leur soutien tout au long de ce stage.

Enfin, j'ai une pensée pour tous mes collègues de travail pour leur aide et la bonne ambiance qu'ils ont pu apporter.

# Introduction

Aujourd'hui, l'accès aux réseaux est de plus en plus ubiquitaire, et on peut voir émerger de nouvelles attentes et de nouveaux comportements des utilisateurs de cette informatique ambiante. On voit également apparaître des modèles dont la caractéristique essentielle est un affranchissement des architectures et infrastructures classiques (réseaux pair-à-pair, ad hoc), et donc de nouveaux problèmes en terme de contrôle et de sécurisation. Les personnes et les systèmes responsables de la gestion des infrastructures réseaux actuelles doivent donc faire face à une complexité grandissante et à une très forte dynamique. Ainsi, la conception, l'évaluation et la gestion de ces réseaux se doit d'intégrer les comportements des utilisateurs et leur impact sur le fonctionnement global.

Les systèmes considérés appartiennent en premier lieu à la catégorie des systèmes complexes. Cette complexité provient avant tout de la diversité des composants, des structures et des interactions mises en jeu. De tels systèmes sont *a priori* ouverts (apparition / disparition dynamique des composants), hétérogènes (morphologie et comportements variés) et formés d'entités composites, distribuées dans l'espace, en nombre variable dans le temps. Ces composants, parmi lesquels l'homme qui, avec son libre arbitre, joue souvent un rôle déterminant, peuvent être structurés en différents niveaux. Les interactions elles-mêmes peuvent être de nature différentes et opérer à différentes échelles spatiales et temporelles.

Le paradigme multi-agents offre un cadre conceptuel permettant la modélisation et la simulation de systèmes complexes faisant intervenir différentes entités en interaction entre elles et avec leur environnement. L'approche multi-agents est ainsi particulièrement adaptée pour la modélisation de systèmes distribués dont les propriétés globales ne découlent pas directement des propriétés des composants du système, comme cela est le cas dans les réseaux pair-à-pair par exemple. Elle permet ainsi d'étudier les conséquences d'un comportement individuel sur la globalité du système.

L'objectif du stage est double. En premier lieu, le but est de modéliser selon le paradigme multi-agents les réseaux pair-à-pair. On verra que cette première problématique en fait apparaître d'autres issues de la modélisation des systèmes complexes et des approches dites multi-modèles. Ensuite, le second objectif est d'étudier l'impact du comportement des utilisateurs sur le fonctionnement du réseau. On s'intéressera particulièrement aux cas de pollution et de partage des ressources : chaque utilisateur étant considéré a priori comme un agent. Des travaux préliminaires ont eu lieu et ont montré l'intérêt de l'approche. Il s'agit ici de considérer et intégrer dans le modèle des caractéristiques réelles quantifiables et d'évaluer leur impact sur le fonctionnement global.

Première partie  
Contexte de l'étude

# Chapitre 1

## Les réseaux Peer-to-Peer (P2P) d'échange de fichiers

### 1.1 Présentation

La notion de P2P n'est pas récente dans le domaine de l'informatique (on peut citer Usenet, un système de diffusion d'information créé en 1979) mais son développement a réellement pris de l'ampleur avec l'arrivée des systèmes d'échange de contenu<sup>1</sup> (fichiers) tel que Napster (1999), Gnutella (2000), eMule (2002), *etc.* Aujourd'hui, environ 80% du trafic d'Internet est généré par cette catégorie [10].

L'échange de ressources informatiques (puissance de calcul, mémoire, contenu) se présente sous la forme de deux modes d'interaction : la connexion dite client-serveur et celle appelée pair-à-pair. Dans le premier cas, le serveur centralise les ressources et les distribue à tous ses clients. Dans le deuxième cas, les utilisateurs qui participent à l'échange sont tour à tour demandeurs et donneurs, clients et serveurs. Ce sont des pairs. L'échange s'effectue donc sur un pied d'égalité.

Pour qu'un réseau P2P existe et fonctionne, il faut tout d'abord que chaque utilisateur installe sur son ordinateur un logiciel qui lui permette de se connecter au réseau (le client P2P). Ensuite, il faut que le réseau soit doté de règles de fonctionnement communes à tous : le protocole du réseau. C'est le rôle du protocole que de définir comment et selon quel format se déroulent les échanges de fichiers. Le réseau, ainsi formé par les utilisateurs, est indépendant du réseau physique sous-jacent (typiquement le réseau IP) et se nomme réseau overlay.

### 1.2 Propriétés

Les propriétés attendues pour les systèmes P2P sont les suivantes.

**L'absence de goulot d'étranglement** Tous les pairs fournissent des ressources (bande passante, mémoire, puissance de calcul). Ainsi, lorsqu'un nouveau nœud rejoint le réseau, la capacité globale du système augmente.

**L'adaptabilité (scalability)** Le système supporte l'ajout ou la suppression dynamique de composants. Les utilisateurs peuvent se connecter ou se déconnecter au réseau sans que cela n'affecte la disponibilité des ressources.

**La robustesse** De part sa nature décentralisée et du fait que chaque pair joue un rôle équivalent, le système est tolérant aux défaillances de ces composants.

---

<sup>1</sup>Dans la suite du document, le terme P2P désignera uniquement les systèmes P2P d'échange de contenu

## 1.3 État de l'art

Aujourd'hui, plusieurs générations de systèmes P2P se sont succédées [9, 25]. De Napster à eMule, cette section se propose d'en faire un rapide aperçu avant de s'attarder sur le protocole étudié durant ce stage : Kademia.

Historiquement, la première génération de réseaux P2P (Napster, Audiogalaxy : 1999) comprenait un serveur central qui maintenait la liste des pairs connectés ainsi que la description des fichiers partagés. Les clients adressaient leurs requêtes au serveur puis l'échange se passait directement entre deux pairs. Mais il ne s'agit pas non plus d'un réseau totalement pair-à-pair dans la mesure où, sans le serveur central, le réseau ne peut pas fonctionner. Cette architecture est qualifiée d'architecture hybride (voir figure 1.1).

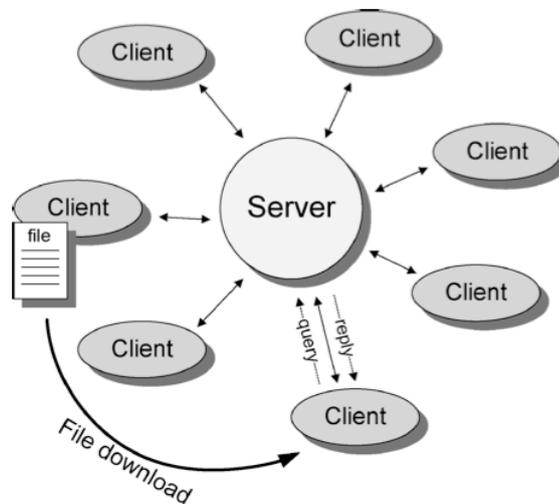


FIG. 1.1 – L'architecture hybride. Les requêtes sont envoyées au serveur central. Les échanges de données se font de pair à pair [9]

Avec la deuxième génération, le serveur central n'est plus nécessaire à la connexion. C'est l'architecture purement décentralisée dans laquelle chaque utilisateur indexe lui-même ses fichiers (voir figure 1.2). Ceux qui sont à la recherche d'un fichier, interrogent, de proche en proche, tous les ordinateurs du réseau. L'exemple le plus connu est le réseau Gnutella version 0.4 (2000). Contrairement aux systèmes de première génération, on a affaire ici à de véritables réseaux pair à pair, puisqu'on peut se passer totalement d'entité centrale. Toutefois, le réseau étant composé de pairs connectés en haut débit et connectés par modem, la vitesse de propagation s'en trouvait limitée par ces derniers.

Ainsi, la troisième génération opère une distinction entre deux niveaux de pairs : ceux qui ont une connexion haut débit et ceux qui ont une connexion par modem. Les ordinateurs disposant d'une connexion par modem se relient à un ordinateur ayant une connexion haut débit. Ce dernier devient dès lors un Super Nœud (SN). Chaque SN indexe alors les fichiers des pairs bas débit qui lui sont rattachés, comme le faisait autrefois le serveur central des réseaux de première génération. Entre deux SN en revanche, le système continue à fonctionner comme les réseaux de seconde génération. Mais la propagation des données est plus rapide, puisqu'elle n'utilise plus que les connexions à haut débit. Une fois l'adresse IP communiquée à l'ordinateur d'origine, une connexion directe s'établit entre les deux pairs, quel que soit leur niveau. Il s'agit donc d'une solution hybride entre les réseaux de première et de seconde génération. C'est l'architecture partiellement décentralisée (Gnutella v0.6, KaZaA 2001).

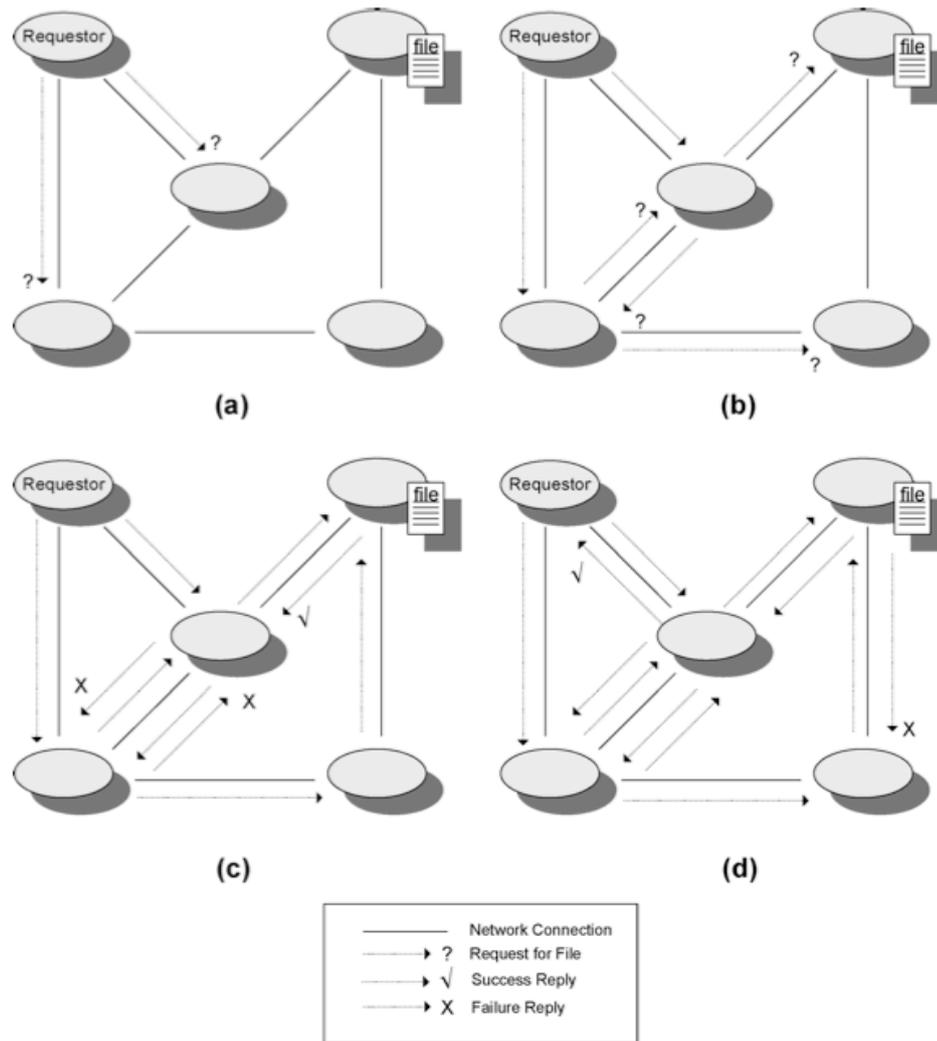


FIG. 1.2 – L'architecture décentralisée. La propagation des requêtes se fait par inondation [9]

Ces trois premières générations sont des réseaux qualifiés de non-structurés. C'est à dire que les liens entre pairs sont établis arbitrairement. De tels systèmes ont l'avantage d'être faciles à créer. Ainsi, lorsqu'un pair veut se connecter, il lui suffit de copier la liste des liens d'un autre pair et de créer sa propre liste au cours du temps. Dans ces réseaux, les requêtes doivent être diffusées de proche en proche. Le principal désavantage est qu'il existe une probabilité que certaines requêtes ne puissent pas être satisfaites même si le fichier existe dans sur le réseau (lorsque l'on cherche des données rares par exemple). La tendance actuelle fait appeler aux réseaux dits structurés.

Les réseaux structurés utilisent un protocole global pour s'assurer que chaque nœud puisse trouver efficacement les ressources qu'il souhaite (même si celles-ci sont rares). Une telle garantie impose de structurer les liens du réseau overlay. La méthode la plus usuelle se base sur l'utilisation de tables de hachage distribuées - en anglais Distributed Hash Table (DHT) - grâce auxquelles chaque pair peut retrouver le propriétaire de chaque fichier. Une des DHT les plus connues est Kademia qui est supportée par le client eMule [1].

	Degré de centralisation		
	Hybride	Partiel	Décentralisé
Non-Structuré	Napster, Publius	KaZaA, Morphéus, Gnutella v0.6	Gnutella v0.4, FreeHaven
Structuré			Chord, CAN, Pastry, Kademia

FIG. 1.3 – Classification résumée des systèmes P2P d'échange de fichiers

**Téléchargement multi-sources** En parallèle de ces évolutions structurelles du réseau overlay, et pour compenser le caractère asymétrique de l'ADSL, des internautes ont eu l'idée de rendre possible le téléchargement à partir de plusieurs sources. Le fichier est *découpé* en petites tranches et peut ensuite être téléchargé morceau par morceau à partir de plusieurs pairs.

## 1.4 Kademia

Kademia est un protocole P2P qui se base sur une table de hachage distribuée. Il fut créé par Petar Maymounkov et David Mazières en 2002 [29]. Il spécifie la structure du réseau overlay et la manière dont se passe l'échange d'information. Les nœuds du réseau Kademia communiquent entre eux via le protocole UDP (User Datagram Protocol).

Chacun de ces nœuds est affilié à un identifiant unique, codé sur 160 bits, appelé *nodeID*. Ces identifiants sont utilisés par Kademia pour rechercher des informations indexées. Ces informations (généralement des mots clefs) sont appelées *value*. Chaque *value* est associée à un identifiant unique (codé là aussi sur 160 bits) appelé *key*. Le principe d'indexation des informations est relativement simple : chaque nœud est responsable d'une liste d'informations - une liste de couple  $\langle key, value \rangle$  - de telle sorte que les identifiants *key* soient proches de son *nodeID*. Ainsi, la distance  $d$  entre deux identifiants  $x$  et  $y$  est définie par la fonction XOR (ou exclusif) entre ces derniers :  $d(x, y) = x \oplus y$ .

Avec cette méthode d'indexation, retrouver une information revient à chercher un nœud dont l'identifiant est proche. Pour ce faire, Kademia considère chaque nœud comme une feuille d'un arbre binaire. La position de chaque nœud est déterminée par le plus petit préfixe unique de son *nodeID*. La figure 1.4 présente un exemple d'arbre dans lequel se situe un nœud dont le préfixe commence par 0011. Pour chaque nœud donné, l'arbre binaire est divisé en une série de sous-arbres qui ne contiennent pas le nœud en question. Le plus grand d'entre eux correspond à la moitié de l'arbre initial (ôté du nœud en question).

Le suivant correspond à la moitié de l'arbre restant et ainsi de suite. Dans l'exemple du nœud 0011, les sous-arbres encadrés sont composés de tous les nœuds dont les préfixes sont respectivement 1, 01, 000 et 0010.

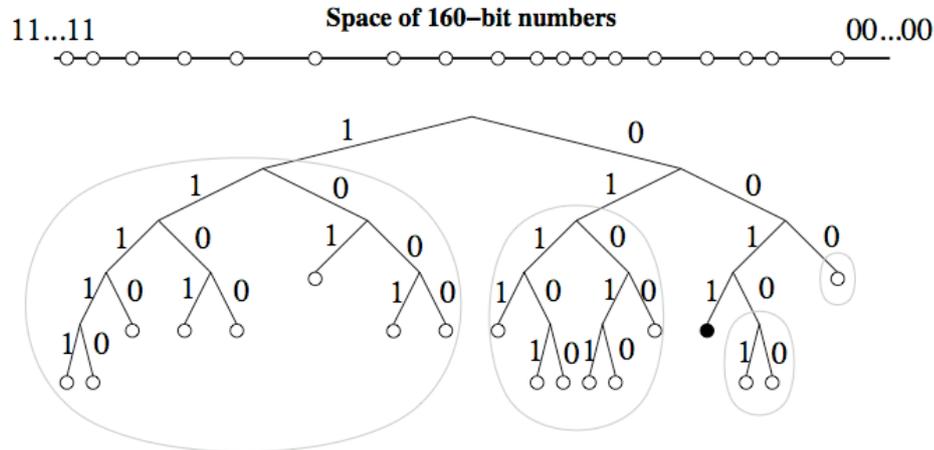


FIG. 1.4 – Arbre binaire Kademlia. Le point noir correspond à l'emplacement du nœud 0011... dans l'arbre. Les ovales gris montre les sous-arbres dans lesquels le nœud 0011... doit avoir des contacts [29]

Le protocole Kademlia s'assure que chaque nœud possède au moins un contact dans chaque sous-arbre. Avec cette garantie, chaque nœud peut trouver tout autre nœud dont l'identifiant est différent du sien. La figure 1.5 apporte l'exemple du nœud 0011 localisant le nœud 1100 en questionnant successivement les meilleurs nœuds qu'il connaît pour trouver des contacts dans les sous-arbres. Finalement l'algorithme converge vers le nœud ciblé.

Le protocole Kademlia utilise quatre procédures - en anglais Remote Procedure Call (RPC) : **Ping**, **Store**, **Find\_node** et **Find\_value**. Chacune prenant en paramètre un identifiant *ID* codé sur 160 bits.

- La procédure **Ping** interroge un nœud pour savoir s'il est connecté au réseau.
- **Store** ordonne à un nœud de stocker un couple  $\langle key, value \rangle$  dans le but de le récupérer ultérieurement.
- **Find\_node** retourne une liste de triplets contenant l'adresse IP, le numéro de port UDP et l'identifiant *nodeID* des nœuds connus pour être les plus proches de l'identifiant ciblé.
- **Find\_value** se comporte de la même manière que **find\_node** avec une exception : si le pair ciblé stocke la valeur (la *value*) recherchée, il la renvoie vers l'initiateur du message.

Dans son implantation réelle (le réseau Kad) le protocole Kademlia est utilisé pour la partie indexation des données. Lors d'un transfert, les clients se connectent directement entre eux via le réseau IP standard. La majorité des clients Kad sont qualifiés d'hybrides. En effet, ces derniers supportent à la fois le réseau Kad et le réseau eDonkey2000 (ed2k). La majorité des utilisateurs du réseaux Kad sont aussi connectés aux serveurs ed2k qui sont utilisés pour trouver des points d'entrée sur le réseau Kad. Selon les estimations [1], il y aurait actuellement entre 3.5 et 5 millions d'utilisateurs sur le réseau Kad pour 500 millions à 1 milliard de fichiers en circulation.

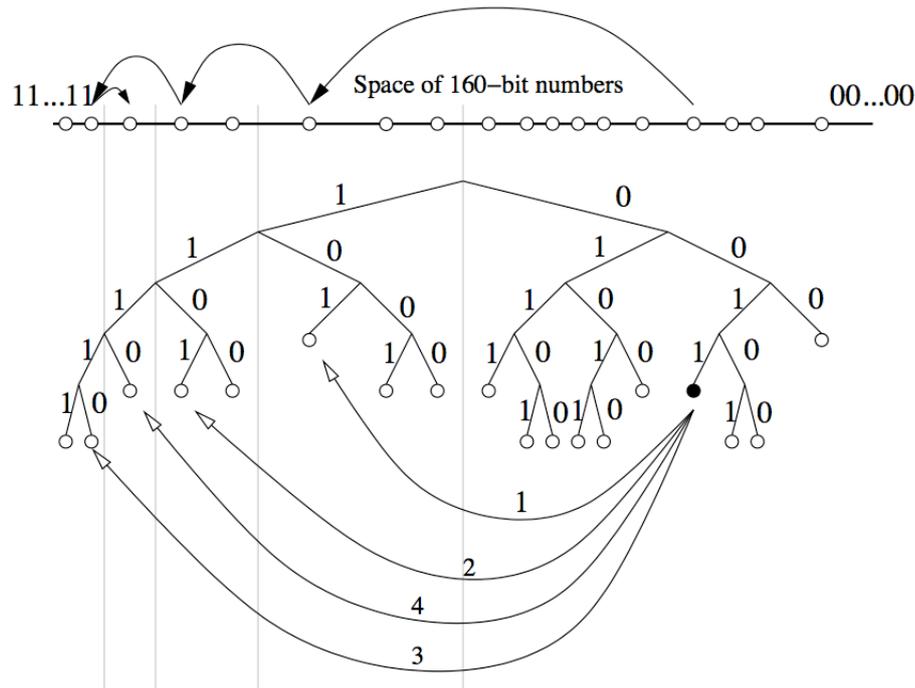


FIG. 1.5 – Localiser un nœud par son identifiant. Ici, le nœud avec le préfixe 0011 trouve le nœuds avec le préfixe 1110 en questionnant successivement les nœuds de plus en plus proches [29]

## 1.5 Impact du comportement des utilisateurs

La topologie, la structure, le degré de centralisation, les méthodes de routages des messages sont autant de caractéristiques capitales pour le fonctionnement du système en terme de performances. Cependant, ces dernières sont généralement dépassées par le comportement de l'utilisateur. En effet, les observations menées sur les réseaux P2P (Napster, KaZaA, Gnutella, *etc.* [8, 36, 24]) ont mis-à-jour des comportements hors-norme et leurs effets pressentis sur les performances globales.

**Les comportements égoïstes** 70 à 80% des utilisateurs tirent des bénéfices des ressources des autres sans partager les leurs en échange (ceci inclut les utilisateurs qui ne veulent pas partager et ceux qui partagent des données que personne ne veut). On comprend intuitivement que la robustesse du système diminue avec l'augmentation du nombre de pairs égoïstes. En effet, les pairs altruistes deviennent d'autant plus sollicités que leur nombre diminue, d'où une augmentation en terme de coût de communication et de calcul. Cela a pour conséquence de les rendre théoriquement plus enclins à tomber en panne et de causer des congestions du réseau. Mais aussi, leur petit nombre les rend plus facilement localisables et donc plus vulnérables aux attaques.

Cette problématique est relativement importante au sein de la communauté P2P. Des mécanismes d'incitations au partage, de révocation ont été proposés ([19] [23]). Le problème, en l'absence de contrôle centralisé, est de faire la différence entre un nouvel utilisateur et un utilisateur malsain qui aurait une nouvelle identité.

**L'empoisonnement des données** Outre la problématique de la diffusion « en clair » de fichiers illégaux (principalement soumis aux droits d'auteur), on voit apparaître une diffusion « cachée » via l'empoisonnement des données. La pollution apparaît dans les réseaux P2P lorsque un large nombre de fichiers cor-

rompu est présent dans le système (Près de 50% des ressources les plus populaires sur le réseau FastTrack [27]). On peut parler de pollution lorsque le nom ou les mots clés sont différents du contenu, ou lorsque le contenu en lui-même est corrompu (virus, spyware, dissimulation d'images, d'information, *etc.*). Avec le téléchargement multi-sources et les réseaux anonymes et cryptés (comme Freenet), l'utilisateur peut être amené à diffuser, à son insu, des ressources corrompues (images pédophiles, propagande extrémiste)[25].

Outre cette problématique juridique et éthique, la présence de pollution au sein d'un système P2P entraîne une réduction de la disponibilité des ressources et une augmentation de trafic. En effet, lorsqu'un utilisateur se rend compte qu'un de ses fichiers téléchargés est corrompu, il a tendance à relancer le téléchargement via une autre source. Un tel empoisonnement du réseau aurait pour conséquence, selon Lee *et al.* [26], de quadrupler, dans le pire des cas, la charge du réseau.

## 1.6 Modélisation et simulation des systèmes P2P

*Les simulateurs de réseaux modernes sont des outils de première importance lorsque l'on désire obtenir des informations relatives aux performances des réseaux et que toute approche plus théorique ne semble pas possible. [...] Dans la majorité des cas, une analyse réaliste des performances des réseaux nécessite la prise en compte de protocoles existants, des équipements existants (routeur, commutateurs, etc.) et des modèles de trafic réalistes. Cette approche descriptive des réseaux ne permet pas la résolution analytique car le nombre de variables, souvent dépendantes du temps, qu'il est nécessaire de prendre en compte est trop grand [12].*

Il existe de nombreux outils pour simuler les réseaux pair-à-pair, depuis les traditionnels simulateurs bas niveau (le plus connu étant certainement NS-2 [2]) jusqu'aux simulateurs plus spécifiques développés récemment dans le domaine du P2P. Cependant, la problématique du comportement de l'utilisateur est relativement nouvelle dans le domaine des réseaux. De fait, elle ne semble être apparue qu'avec les réseaux P2P. Jusqu'à présent, les recherches dans le domaine du P2P se sont principalement concentrées sur l'optimisation des protocoles en faisant l'hypothèse d'utilisateurs dénués d'intelligence, qui ne font que suivre un protocole établi. Ainsi, les outils de simulation des réseaux P2P présentent rarement une couche « modélisation de l'utilisateur » (voir section 5.2.0.0.0).

De plus, les études déjà menées sur la problématique du comportement des utilisateurs ([18, 26, 13]) prennent rarement en compte les caractéristiques réseau comme par exemple les temps de latence ou de téléchargement, ni l'hétérogénéité spatio-temporelle des utilisateurs (comme le fait qu'environ 80% des utilisateurs restent connectés moins de trois heures [36]), ni même la popularité des fichiers. Pourtant ces caractéristiques ne semblent pas pouvoir être séparées de la problématique. En effet, même si le comportement semble en grande partie issue des caractéristiques intrinsèques des individus (telles que la volonté de partager ou la conscience de la pollution) il est aussi influencé par la perception qu'a l'utilisateur de son « environnement P2P » (*i.e* la bande passante consommée par le client P2P, le temps de téléchargement, *etc.*). Par exemple, si la demande en ressource semble trop élevée, l'utilisateur aura tendance à diminuer son offre voire à ne plus partager du tout. De plus, il semble important, pour pouvoir mesurer l'impact du comportement, de disposer d'une modélisation relativement réaliste du réseau sous-jacent et ainsi généraliser les résultats obtenus.

À l'opposé, le paradigme multi-agents permet d'étudier les conséquences d'un comportement individuel sur la globalité du système [17, 20, 28]. Cette approche offre un cadre conceptuel permettant la modélisation et la simulation de systèmes complexes faisant intervenir différentes entités en interaction entre elles et avec leur environnement. Elle est ainsi particulièrement adaptée pour la modélisation de systèmes distribués dont les propriétés globales ne découlent pas directement des propriétés des composants du système, comme cela est le cas dans les réseaux P2P.

## 1.7 Synthèse

Dans cette première partie, on présente rapidement les systèmes P2P, leur propriétés et leur fonctionnement. Ensuite, les problématiques de coopération (partage des fichiers) et de pollution et le rôle qui y est joué par les utilisateurs sont présentés. Enfin, on aborde la problématique de la modélisation et de la simulation des systèmes P2P lorsque l'on s'intéresse à l'impact du comportement des utilisateurs dans de tels systèmes. On apporte un élément de réponse en évoquant la possibilité d'utiliser le paradigme multi-agents dans la modélisation et la simulation des réseaux P2P. Le chapitre suivant présente l'approche par systèmes multi-agents.

## Chapitre 2

# Un mot sur l'approche par Système Multi-Agents (SMA)

*The use of agent systems to simulate real-world domains may provide answers to complex physical or social problems that would otherwise be unobtainable due to the complexity involved, as in the modelling of the impact of climate change on biological populations, or modelling the impact of public policy options on social or economic behaviour. [...] Multi-agent simulation models thus effectively provide a new set of tools for the management of complex adaptive systems, such as large-scale online resource allocation environments (Luck et al. [28]).*

## 2.1 Notion d'agent

### 2.1.1 Définition

On voit apparaître la notion d'agent autonome dans la littérature au cours des années 1970 (Lenat - 1975, Hewitt 1977, Minsky - 1978) mais le développement des systèmes à base d'agents n'émergera que plus tard, au cours des années 1990. Une des définitions généralement acceptée dans le domaine est celle énoncée par Ferber en 1995 [20] : *On appelle agent une entité informatique qui :*

1. *se trouve dans un système informatique ouvert comprenant un ensemble d'applications, de réseaux et de systèmes hétérogènes,*
2. *peut communiquer avec d'autres agents,*
3. *est mue par un ensemble d'objectifs propres,*
4. *possède des ressources propres,*
5. *ne dispose que d'une représentation partielle des autres agents,*
6. *possède des compétences qu'elle peut offrir aux autres agents,*
7. *a un comportement tendant à satisfaire ses objectifs, en tenant compte, d'une part, de ces ressources et des compétences dont elle dispose et, d'autre part, de ces propres représentations et des communications qu'elle reçoit.*

### 2.1.2 Propriétés

D'après cette définition, on peut extraire quelques grandes propriétés que doivent satisfaire les agents.

- Ils sont autonomes : les agents ont la possibilité d'agir sans intervention directe humaine. On dit aussi qu'ils sont proactifs.
- Ils sont flexibles : ils peuvent réagir aux changements de l'environnement, s'adapter aux ressources disponibles.
- Ils sont sociaux : ils ont la capacité d'interagir pour atteindre leurs buts, pour aider d'autres agents dans leur activité.

- Ils sont situés : les agents ont la capacité de percevoir l’environnement et à y agir de façon limitée.

## 2.2 SMA et applications

L’approche des SMA composés d’agents réactifs met l’accent sur l’émergence de comportements collectifs évolués (voire complexes) à partir de comportements individuels élémentaires, cette émergence étant le fruit des interactions des agents entre eux. Si l’on désire étudier l’influence de ces interactions sur le fonctionnement général, c’est donc vers de tels systèmes qu’il faut se tourner.

On pourrait résumer la notion de SMA à un système (distribué) composé d’un environnement où peuvent évoluer des ensembles organisés d’agents en interaction. Plus formellement, un cadre de conception des SMA fut défini par Demazeau en 1995 [15]. C’est l’approche A.E.I.O. ou encore approche voyelle. Elle se base sur quatre briques élémentaires :

- A comme Agent.
- E comme Environnement. Milieu dans lequel évoluent et interagissent les agents.
- I comme Interactions. Infrastructures, langages et protocoles d’interaction entre agents, depuis de simples interactions physiques jusqu’aux actes de langages.
- O comme Organisation. Hiérarchies, structures et relations entre les agents.

Les propriétés attendues des SMA sont généralement les suivantes. De part sa nature décentralisée, le système est ouvert et robuste. C’est-à-dire que l’on peut dynamiquement ajouter ou enlever des agents (du fait des besoins ou de défaillances) sans remettre en cause l’aboutissement de l’objectif global. De plus, le système s’adapte, en temps réel, aux changements d’environnement (baisse de ressources par exemple) [33].

*Le champ d’application des systèmes multi-agents est actuellement en pleine expansion. Il, semble, en effet que les domaines dans lesquels les SMA vont jouer un rôle crucial dans les années à venir concernent : l’intelligence ambiante, où la problématique principale est de fournir en tout lieu à l’utilisateur interactivité et réactivité ; le Grid Computing, où la problématique concerne principalement l’utilisation efficace des capacités de calcul mises à disposition dans le monde scientifique ; le commerce électronique, où l’objectif est d’utiliser ces technologies pour (semi)automatiser les procédures commerciales ; le Web Sémantique, où les agents fournissent l’infrastructure nécessaire à la manipulation de la connaissance mise à disposition... [33].*

## 2.3 Agents réactifs et cognitifs

Dans le domaine des SMA, une classification des types d’agents est généralement faite selon la complexité du processus de décision. La plus couramment acceptée est la distinction entre agents cognitifs et réactifs. Les agents cognitifs sont inspirés des sciences sociales, possèdent une représentation complexe de leur environnement et des capacités de planification et de communication sophistiquées. En général leur nombre est relativement faible.

Par opposition, la métaphore sous-jacente des agents dits réactifs est inspirée de la biologie (colonies de fourmis, bancs de poissons, etc). Ces agents n’ont pas de représentation interne de leurs propres compétences, et n’ont qu’une représentation très succincte de leur environnement sous la forme de quelques indices numériques ou logiques, directement issus de leur phase de perception. Ils ne sont pas censés élaborer de plan ni même choisir une stratégie mais se contentent au contraire d’appliquer des traitements préétablis en réponse à un ensemble de stimulations attendues (processus de type stimulus-réponse).

## 2.4 La simulation multi-agents

À l'inverse des approches de modélisation basées sur la définition d'équations mathématiques, où la dynamique du système est définie dès le début par des relations fonctionnelles entre entités, la simulation multi-agents se propose de modéliser explicitement les comportements des entités et considère que la dynamique globale d'un système, au niveau macroscopique, résulte directement de l'interaction des comportements, au niveau microscopique (Parunak *et al.* [17]).

L'intérêt de ces simulations est de pouvoir considérer aussi bien des paramètres quantitatifs (c'est-à-dire des paramètres numériques) que qualitatifs (des comportements individuels faisant appel à des raisonnements stratégiques). Ce sont des constructions de *micro-mondes artificiels* dont il est possible de contrôler toutes les caractéristiques et de reproduire des séries d'expérimentations comme s'il s'agissait de simulations de laboratoire (Ferber [20]).

La simulation multi-agents est particulièrement bien adaptée lorsque les domaines présentent les caractéristiques suivantes :

- Les entités présentes dans le système peuvent aussi bien être dispersées géographiquement que temporellement.
- Lorsque les entités présentent une autonomie de décision.
- Lorsque l'environnement est évolutif voire incertain.
- Lorsqu'une approche centralisée plus « classique » fait défaut.

Ainsi la simulation multi-agents est utilisée pour l'étude de phénomènes complexes réels ou virtuels dans les domaines tels que l'éthologie, la sociologie, l'économie, l'environnement, l'animation, *etc.* Elle permet de représenter et de simuler des systèmes faisant intervenir un grand nombre d'individus (écosystèmes, sociétés d'insectes, dynamiques de populations [35]), de modéliser des environnements virtuels (jeux-vidéo, cinéma [3]). Enfin, elle permet d'expliquer l'impact de comportements individuels à un niveau global de description.

La complexité d'échelle, tant aussi bien quantitative que spatiale et temporelle, des réseaux P2P nous renforce dans l'idée que la modélisation et la simulation des systèmes P2P, du moins pour cette problématique d'impact du comportement, passe par une approche multi-agents. D'ailleurs, des travaux ont déjà été effectués à la croisée des deux domaines (routage, diffusion et recherche d'information, confiance et réputation, *etc* voir [21] et [30]). Néanmoins, il existe relativement peu d'études sur la simulation multi-agents des réseaux P2P. La plupart se concentrent sur l'introduction d'agents mobiles à l'intérieur même du réseau afin d'automatiser certaines tâches (voir Anthill [11], MATES [38]).

## 2.5 Synthèse

Cette partie offre une introduction aux SMA. On présente ici la notion d'agent, de système multi-agents et de simulation multi-agents. On remarque aussi que les propriétés attendues d'un SMA sont relativement proches de celles des réseaux P2P. Ainsi, on se conforte dans l'idée qu'un rapprochement entre les deux domaines est réalisable. Cependant, on découvre que relativement peu de travaux se sont penchés sur la problématique de la simulation des réseaux P2P par une approche multi-agents.

# Chapitre 3

## Présentation des objectifs

### 3.1 Synthèse des problématiques

Nous avons vu précédemment les différentes problématiques issues des comportements des utilisateurs (coopération et pollution). Nous avons aussi montré que, malgré les optimisations structurelles des protocoles P2P, il semblait nécessaire de prendre en compte le comportement de l'utilisateur dans les modélisations et les simulations. Après avoir considéré les différentes études portant sur cette question, il nous est apparu que certains points pouvaient être discutés : lors des études et lors de la conception des systèmes P2P, il est rarement fait cas de l'hétérogénéité des comportements des utilisateurs (bien souvent ces derniers suivent un protocole établi). À l'opposé, lorsque l'on s'intéresse au comportement des utilisateurs, c'est la modélisation des réseaux sous-jacents qui fait défaut.

### 3.2 Objectifs

Ainsi, on souhaite proposer une modélisation des systèmes P2P qui tienne compte des comportements des utilisateurs, de leur hétérogénéité spatiale et temporelle et qui inclut des caractéristiques du réseaux sous-jacent permettant d'évaluer l'impact du comportement des utilisateurs sur le fonctionnement du réseau. Ensuite, on souhaite proposer un cas d'étude - l'impact réciproque du partage et de la pollution des données sur le fonctionnement du réseau - qui permettra à la fois de valider cette approche conceptuelle et d'avancer sur la question de l'impact du comportement des utilisateurs dans les réseaux P2P.

La suite du mémoire s'organise de la façon suivante : tout d'abord, nous proposerons une modélisation des systèmes P2P s'appuyant sur une approche multi-agents. Nous montrerons ensuite les différentes problématiques qui en découlent. Enfin, nous présenterons un cas d'étude qui nous permettra d'avancer dans cette direction et d'évaluer notre démarche.

## Deuxième partie

# De la modélisation à la simulation des réseaux P2P via une approche multi-agents

## Chapitre 4

# P2P et SMA : quels besoins en terme de simulation ?

### 4.1 Analogie

On peut facilement faire le lien entre les SMA et les réseaux P2P. La première caractéristique qui vient à l'esprit lorsqu'on les compare est leur nature intrinsèquement distribuée ainsi que l'absence de contrôle centralisé. Ce sont tous deux des systèmes distribués qui font intervenir des entités indépendantes (agents ou pairs) n'ayant qu'une vision limitée de leur environnement et des autres entités, et qui échangent des messages et services au travers d'un environnement donné (ici, Internet).

Si on reprend la définition d'un agent donnée par Ferber (voir section 2.1.1.0.0) et que l'on remplace simplement le mot *agent* par *pair*, on obtient une définition étrangement conforme à celle du pair dans les systèmes P2P. De ce point de vue, le parallèle entre SMA et réseaux P2P semble se faire intuitivement. Si on reprend le formalisme de Demazeau, l'approche voyelle, pour décrire un réseau P2P, on peut le voir de la manière suivante :

- Les agents correspondent aux pairs du réseau.
- L'environnement au réseau IP.
- Les interactions entre agents et leur organisation peuvent être vus comme le protocole P2P.

La figure 4.1 résume cet état d'esprit.

### 4.2 Une approche multi-modèles

Cette démarche fait clairement entrevoir une problématique au niveau modélisation. En effet, on observe deux niveaux d'abstraction principaux : un niveau « réseau » (qui peut être lui même divisé en une partie overlay et une partie IP) et un niveau « comportement de l'utilisateur » dont les besoins, en terme de modélisation, sont assurément différents. On peut citer, en plus d'avoir affaire à deux domaines distincts (voir les sections suivantes), par exemple les différences en terme de gestion du temps. D'une manière générale, on peut se demander comment faire interagir deux modèles dont les échelles temporelles et la représentation même du temps diffèrent. Cette problématique, seulement évoquée dans cet exposé, est à l'heure actuelle encore le sujet de recherches (voir [16] pages 30-33). Cependant, en utilisant un modèle d'agent réactif, dont le comportement est relativement simple, on peut éviter, dans un premier temps, certaines de ces questions.

La modélisation du comportement des utilisateurs des réseaux P2P fait intervenir des domaines comme les sciences cognitives ou l'économie. Les études déjà réalisées font état de différents modèles. La problématique qui se pose ici est comment peut on les appliqués via le formalisme établi en SMA. Pour ce faire, il est nécessaire de tenir compte des contraintes imposées par le SMA comme, par exemple, la

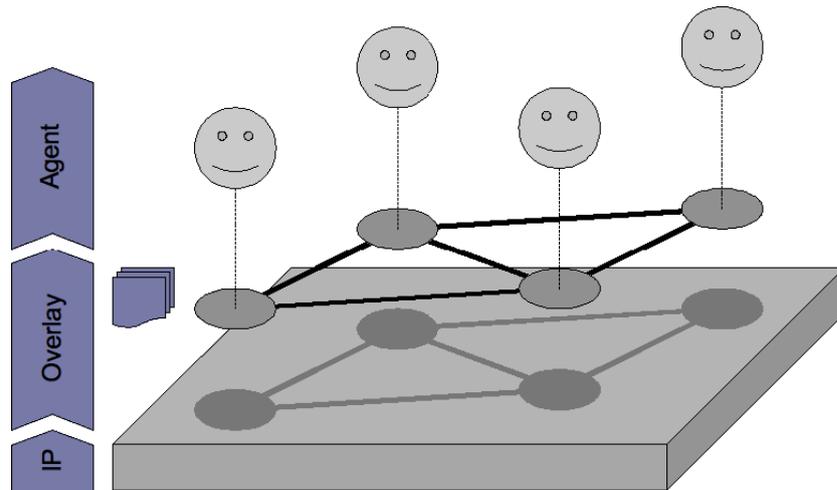


FIG. 4.1 – Modélisation du système P2P via le paradigme multi-agents

perception locale de l'agent : à aucun moment l'agent ne peut avoir une vue globale de tout le réseau P2P. La section suivante propose une modélisation possible du comportement d'un utilisateur transposée dans le paradigme multi-agent.

### 4.3 Modélisation de l'agent

Le nombre d'agents que l'on souhaite intégrer au système est relativement élevé (entre  $10^3$  et  $10^6$ ). De plus, la représentation qu'a un utilisateur de ces « voisins » et du fonctionnement du réseau peut-être simplifiée. En effet, on peut considérer que l'utilisateur ne peut percevoir et interagir sur le réseau qu'au travers de son client P2P. C'est, en outre, pour ces deux raisons que l'on se propose d'utiliser des agents réactifs. Le comportement des agents réactifs est déterminé par un cycle perceptions - actions. L'agent perçoit certains états de l'environnement via ses capteurs, et décide des actions à mener en fonctions de ses percepts et de ses états internes. Dans notre cas, les perceptions sont données par la vue qu'offre le client P2P et l'ensemble des actions possibles est défini par l'ensemble des actions permises par le client P2P.

On voit ici apparaître le rôle que joue le client P2P en tant qu'interface entre le modèle du comportement de l'utilisateur et le réseau. C'est par lui que transitent toutes les interactions entre les deux mondes. Pour l'utilisateur, le client cache les aspects réseaux (protocole, type de messages), c'est une sorte de fenêtre sur le monde au travers de laquelle il ne perçoit que des sources de données et qui lui permet simplement de chercher une ressource et de la télécharger. À l'inverse, au niveau réseau, le client apparaît comme l'émetteur et la destination des différents messages (`ping`, `find_value`...). La section suivante présente les difficultés liées au besoin de modélisation du deuxième modèle : l'environnement P2P.

### 4.4 Modélisation de l'environnement

Au niveau de la modélisation du réseau, il apparaît que celle-ci ne doit pas obligatoirement être trop détaillée. En effet, du fait du nombre important d'utilisateurs il ne semble pas nécessaire de modéliser jusqu'à l'envoi de chaque datagramme sur le réseau, voire de modéliser l'ensemble des équipements réseaux (routeurs, commutateurs *etc*). L'envoi de messages au niveau overlay s'avère être un bon compromis entre la taille du réseau et la complexité des interactions entre les pairs. Cependant, il nous paraît nécessaire de prendre en compte quelques caractéristiques du réseau sous-jacent comme le temps de transfert d'un

message d'un utilisateur à un autre (temps de latence) ou encore le temps que prend l'échange de données entre deux pairs (temps de téléchargement).

## 4.5 Synthèse

Cette partie évoque les différentes problématiques issues de la modélisation des réseaux P2P par une approche orientée agents. Ainsi, après avoir montré la forte analogie qui existe entre P2P et SMA, on évoque la problématique de multi-modélisation et le rôle que tiens l'interface entre le niveau agent (les utilisateurs) et le niveau environnement (le réseaux).

Il existe un réel manque au sein de la communauté multi-agent en terme d'outils de modélisation et de simulation des réseaux P2P. En effet, même si les réseaux sociaux (ensemble de relations entre individus) peuvent être modélisés et simulés relativement facilement via les plate-formes existantes (comme MadKit [4], Repast [6] voire PeerSim [7]), ce niveau d'abstraction élevé ne convient pas, à lui seul, à la problématique « d'impact sur le fonctionnement des réseaux P2P ». La nécessité de prendre en compte un environnement P2P cohérent (un protocole P2P existant et une modélisation du réseau sous-jacent correcte), le manque de simulateurs multi-agents adaptés et la contrainte de temps ont orienté le choix d'un outil de simulation vers les simulateurs P2P existants. La section suivante décrit comment s'est fait ce choix et quel outil ressort parmi les différents analysés.

# Chapitre 5

## Choix d'un outil de simulation

### 5.1 Critères utilisés

Cette section n'a pas pour prétention d'établir une liste exhaustive des outils de simulation des systèmes P2P, mais plutôt de déterminer, d'après une série de critères pertinents vis-à-vis de notre problème, les simulateurs qui s'y conforment.

**Niveaux de détail** Comment sont modélisés / simulés (si ils le sont) le(s) comportement(s) des utilisateurs, le(s) protocole(s) pair-à-pair et le réseau physique? En effet, on a vu dans la section précédente la nécessité de prendre en compte les différents niveaux d'abstraction.

**Le passage à l'échelle** Quel est le maximum de noeuds que le simulateur peut gérer? Est-ce annoncé et / ou vérifié? Et dans quelles circonstances? Le nombre d'utilisateurs des systèmes pair-à-pair étant relativement élevé (de l'ordre de  $10^6$ ), il est nécessaire, pour des raisons de validité, que notre simulateur puisse, en un temps raisonnable, effectuer une simulation sur un nombre conséquent de pairs.

**La gestion du temps** La simulation est-elle cyclique ou guidée par les événements? Les systèmes réels que l'on étudie sont souvent continus dans le temps. Cependant, les observations que l'on en fait, se portent fréquemment sur des instants bien précis (ici, l'envoi d'un message, un temps d'attente écoulé, *etc*). Il y a donc une discrétisation qui s'opère entre la réalité et la simulation. Les outils de simulation des réseaux gère le temps principalement de deux manières : soit la simulation est cyclique (on regarde l'état du système à intervalles fixes) ; soit la simulation est guidée par les événements (un échéancier gère les événements qui doivent survenir). Par souci de réalisme, les réseaux P2P étant fortement asynchrones, on privilégiera la simulation guidée par les événements.

**Mesures et Statistiques** Le simulateur génère-t-il directement des statistiques et quels sont les types de mesures que l'on peut effectuer? Quels types (espérance, variance, intervalles de confiance, *etc*)? Comment peut-on les traiter? Le cas échéant, peut-on intégrer facilement un outil de statistiques extérieur? Les processus intervenant dans les réseaux et systèmes pair-à-pair sont rarement déterministes (pannes, perte de messages, paquets...). Ainsi, afin d'analyser au mieux les résultats des simulations, les outils de statistiques sont les bienvenus.

**Activité du projet** Le simulateur est-il en cours développement? soutenu par une communauté? utilisé lors de recherches? Ce critère est important dans la mesure où les expériences et résultats doivent être reproductibles afin d'être validés par la communauté scientifique.

**Prise en main** L'apprentissage du fonctionnement du logiciel (cela inclut le langage de programmation utilisé), la création d'entrées pour le simulateur et l'obtention de résultats sont-ils aisés? Peut-on facilement créer un modèle d'utilisateur dans ce logiciel (cela inclut aussi la notion d'*open source*)? Le

logiciel est-il documenté ? Cette documentation, si elle existe, est-elle abondante ? Le temps imparti pour cette étude n'étant pas forcément très long, l'idéal serait d'utiliser un logiciel de simulation dont la courbe d'apprentissage ne soit pas trop forte.

## 5.2 Les Simulateurs pair-à-pair existants

Le tableau 5.1 présente un résumé de la revue bibliographique des différents simulateurs P2P. Une remarque que l'on pourrait soulever ici, est l'absence d'un simulateur P2P « standard » [32]. Malgré des efforts dans ce sens et à l'inverse de la simulation des réseaux dits « bas niveau » dont la référence est NS-2 [2], la simulation P2P ne semble pas avoir atteint un stade de maturité satisfaisant pour pouvoir bénéficier d'un référent reconnu par tous. La section suivante présente le simulateur qui semble le plus correspondre aux critères énoncés ci-dessus : Peerfactsim.KOM [5].

Simulateur	Avantages	Inconvénients
P2PSim [31]		Statistiques limitées, passage à l'échelle : 3000 noeuds
PeerSim [7]	passage à l'échelle : $10^6$ noeuds en mode cyclique	Réseau physique non modélisé, Peu de documentation sur le mode événement discret
Narse [31]		passage à l'échelle : 600 noeuds
QueryCycle [37]	Spécialisé dans les systèmes d'échange de fichiers, modélisation du réseau	mode cyclique, passage à l'échelle : 20 noeuds
PlanetSim [22]	passage à l'échelle : $10^5$ noeuds	pas de statistiques, réseau physique non modélisé
NeuroGrid [31]	passage à l'échelle : $3 \times 10^5$ noeuds	Réseau physique non modélisé, statistiques limités
GPS [31]		relativement jeune donc peu de feedback
3LS [39]	modèle du système, visualisation du réseau et statistiques	passage à l'échelle : <1000 noeuds
P2PRealm [31]		Spécifique aux réseaux de neurones
OverSim [31]	passage à l'échelle $10^5$ noeuds, réseau physique modélisé, statistiques	protocoles implantés : Chord et GIA

FIG. 5.1 – Caractéristiques techniques

## 5.3 Peerfactsim (PFS)

PeerfactSim.KOM [5] est un simulateur P2P à événements discrets écrit en Java, développé par l'université technique de Darmstadt (Technische Universität Darmstadt) et publié sous licence GNU General Public Licence. Le but affiché par les initiateurs du projet est de créer un outil de simulation adaptatif et léger permettant une modélisation efficace, précise et réaliste des protocoles et applications P2P.

**Avantages** Ce simulateur répond particulièrement à nos attentes dans le sens où son architecture fait nettement la séparation entre les différents niveaux d'abstraction décrits précédemment (voir 5.3.0.0.0) :

- Le niveau réseau IP, le simulateur modélise les temps de latence pour chaque message envoyé.
- Le niveau réseau overlay, des protocoles P2P existants (CAN, Omicron, Chord, Kademia) sont implantés originellement dans le simulateur.

- Le niveau utilisateur, la partie comportement de l'utilisateur est totalement détachée du nœud physique correspondant. Ainsi, on peut implanter un modèle de comportement sans avoir à modifier les parties réseaux et introduire un biais dans la simulation.

Ensuite, la répartition spatiale et temporelle des utilisateurs est prise en compte. Pour mémoire, c'est une des problématique qui avait influencer le choix de l'approche par SMA. Enfin, le nombre de nœuds énoncé est de l'ordre de  $10^5$ , sur un réseau Omicron [14], une valeur intéressante au regard des critères énoncés et des performances annoncées des autres simulateurs.

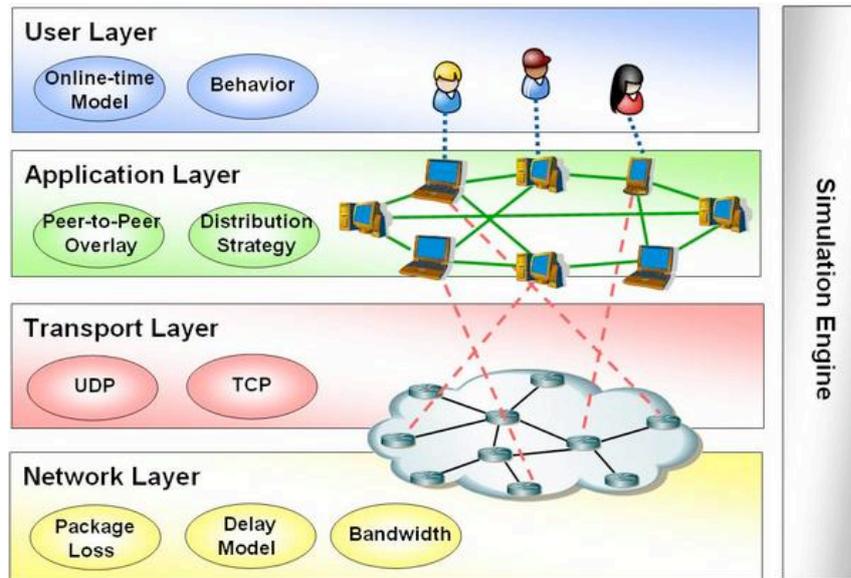


FIG. 5.2 – Architecture de PFS. [5]

**Inconvénients** La documentation est relativement faible et le seul moyen d'apprendre le fonctionnement du simulateur est de se plonger dans son implantation et d'avoir une certaine connaissance des protocoles P2P qui y sont implantés. De plus, du fait de sa relative jeunesse, les premières publications impliquant PeerfactSim sont peu nombreuses et remontent à 2005. Le simulateur n'est pas encore aussi complet que prévu et continue d'être développé. On peut citer notamment la modélisation de la bande passante, ou encore l'implantation du protocole gnutella qui ne sont, pour le moment, pas encore finalisées.

## 5.4 Synthèse

Cette partie présente les différents critères de sélection utilisé pour le choix de notre simulateur. Ceux-ci sont, entre autre, issus des différentes remarques émises jusqu'à présent. Le simulateur PeerfactSim.KOM y est aussi présenté, ainsi que ses avantages et ses inconvénients vis-à-vis de notre problématique.

# Chapitre 6

## Présentation du cas d'étude

### 6.1 Problématique

**Pollution vs Coopération** Ces deux notions, égoïsme et pollution, semblent, d'une certaine manière, en contradiction. En effet, pour que le système fonctionne convenablement, il apparaît nécessaire de développer la coopération (plus précisément le partage). Mais celle-ci, en présence de pollution, implique une diffusion plus rapide des ressources corrompues et donc une baisse significative des performances globales : trafic important, baisse de la coopération, baisse de la disponibilité des ressources. L'étude réalisée par Lee *et al.* semble établir que, lorsque le taux de pollution est important, la conscience qu'a chaque utilisateur de la pollution prévaut sur le partage des fichiers. Plus largement, on peut se demander si la présence d'égoïstes ne protégerait pas, d'une certaine manière, le système contre les pollueurs ?

On se propose donc de se positionner dans ce cas d'étude pour pouvoir développer un modèle précis, l'implanter au sein du simulateur PFS et, d'une part, valider notre approche et, d'autre part, pouvoir répondre en partie à la problématique de l'impact du comportement des utilisateurs sur le fonctionnement des réseaux P2P.

### 6.2 Définition des besoins

Étudier le partage et la pollution au sein des réseaux P2P implique de définir et de modéliser ces deux notions. Tout d'abord, nous nous proposerons de modéliser le partage c'est à dire définir les critères selon lesquels un utilisateur partage ou non ses données. Ensuite, à partir de la définition de la pollution, nous présenterons une modélisation de la pollution au sein des fichiers.

Troisième partie

Contributions

# Chapitre 7

## Modélisation du comportement de l'utilisateur

### 7.1 Un modèle d'agent réactif

Nous avons présenté précédemment les raisons qui nous ont poussé à faire le choix de l'utilisation d'agents réactifs. Cette partie présente la modélisation des utilisateurs des réseaux P2P via l'architecture d'agent réactif dit « à états internes » proposée dans le cadre de notre cas d'étude. Pour rappel, le comportement des agents réactifs est déterminé par un cycle perceptions - actions. L'agent perçoit certains états de l'environnement via ses capteurs (ici le client P2P), et décide des actions à mener en fonctions de ses percepts et de ses états internes. Après avoir examiné les différentes études se rapportant à la question de la pollution et de la coopération ([19, 26, 34]), on se propose de considérer l'ensemble des paramètres suivant comme un ensemble minimum (toujours dans le cadre coopération / pollution) pour modéliser le comportement des utilisateurs.

#### 7.1.1 Perceptions

Les perceptions sont données par la vue qu'offre le client P2P. On considérera les suivantes :

- Le débit descendant (download), la bande passante consommée par le téléchargement de ressources.
- Le débit montant (upload), la bande passante consommée par l'offre de ressource. On peut considérer le rapport entre ces deux débits comme le coût de la contribution de l'agent comme évoqué dans [19].
- Nombre de sources disponibles après un certain temps d'attente.
- Temps de téléchargement estimé.

#### 7.1.2 Actions

On peut aussi définir l'ensemble des actions possibles comme l'ensemble des actions permises par le client P2P utilisé. La première action étant, bien entendu, la connexion / déconnexion au réseau P2P. Dans cette étude, on considérera, de surcroît, les actions suivantes :

- Envoie d'une requête, recherche d'une ressource donnée.
- Lancement de la procédure de téléchargement.
- Arrêt de la procédure de téléchargement (lorsque celui-ci n'est pas terminé).
- Partage ou non des ressources (ou d'une certaine quantité).
- Contrôle du taux de pollution de certaines données.

#### 7.1.3 États internes

L'ensemble des états internes de l'agent, dans notre cas, peut être défini par les états suivants :

- La volonté de partagé (générosité  $\alpha$ ). Cette notion est issue de l'économie et des travaux de Andreoni sur le modèle *warm-glow model* (voir l'utilisation faite dans [19]). On peut considérer cet état comme la quantité (en pourcentage) de ressource que l'utilisateur est prêt à partager.
- La conscience de la pollution  $\beta$ , évoqué dans [26]. On peut voir cet état comme la probabilité que l'utilisateur détecte la pollution sachant que le fichier est corrompu. De cette manière, on peut dire qu'un agent dont la conscience  $\beta = 1$  va automatiquement détecter un fichier pollué et va agir en conséquent (dans notre cas, l'utilisateur supprime la donnée polluée). À l'opposé, un utilisateur dont la conscience  $\beta = 0$  va automatiquement laisser passer les fichiers pollués et s'il les partage, propage cette pollution. De plus, on fait l'hypothèse que l'utilisateur reconnaît toujours un fichier authentique.
- L'intérêt pour le système (besoin de ressources  $\gamma$ ). On peut imaginer, de la même manière que dans [34], que l'utilisateur, s'il est obligé de redemander des ressources ou si le temps de téléchargement est trop important, perd de l'intérêt à rester connecté. Ainsi, si son intérêt  $\gamma$  passe sous une valeur seuil, il pourrait se déconnecter et réintégrer le système plus tard.

## 7.2 Synthèse

Ce modèle ainsi posé nous permet d'établir une base sur laquelle différents schéma comportementaux pourront être développés en fonction des études menées (voir la section 9.1). Pour en revenir à la problématique multi-modèle, on voit ici apparaître le rôle de l'interface entre l'environnement et l'agent (le client P2P) en ce qui concerne les échanges entre les deux modèles. Le chapitre suivant évoque les différentes modifications qu'il a fallu faire subir à cet outil pour obtenir les bases d'un simulateur multi-agents consacré aux réseaux P2P.

## Chapitre 8

# Modification effectuées au sein du simulateur

### 8.1 Du simulateur P2P au laboratoire virtuel

Malgré une architecture proche de notre souhait, et du fait de la jeunesse du projet, le simulateur en tant que tel ne pouvait pas nous satisfaire. PeerfactSim est avant tout un simulateur P2P avant d'être un *framework* plus général sur le P2P (but à terme, voir [5]). C'est à dire que la simulation est focalisée sur les échanges de messages du niveau overlay. Ainsi, pour pouvoir étudier l'impact des utilisateurs égoïstes et celui de la présence de pollution dans le système (au sein d'un réseau Kademlia), il a fallu modifier le simulateur. Remarque : le simulateur étant *open-source* (sous licence GNU-GPL), le code et les commentaires implantés sont rédigés en anglais.

#### 8.1.1 Le partage des fichiers

Le phénomène de coopération (*i.e.* le partage des données) est une problématique essentielle dans les réseaux P2P. Elle peut être décrit de la manière suivante :

- La première étape consiste à publier des données. Posons  $P$  le nombre d'utilisateurs qui publient au temps  $t = 0$  un fichier  $f$ .
- Au bout d'un certain intervalle de temps  $\delta$ , posons  $M$  le nombre d'utilisateurs qui ont téléchargé ce fichier  $f$ . Si ces  $M$  personnes partagent leurs ressources, on peut espérer que le nombre de sources de téléchargement au temps  $t = \delta$  est au mieux de  $P + M$ .

Ainsi d'après ce procédé, une des motivations originelles du P2P, la disponibilité d'une ressource est censée croître avec le nombre d'utilisateurs coopérant. Cependant, ce comportement est idéal et, on l'a vu précédemment, on observe bien souvent des comportements égoïstes.

Cette notion de fichiers partagés (ou non) n'était pas présente dans la version du simulateur que nous possédions. C'est à dire que chaque pair n'avait pas d'endroit concret où stocker ses fichiers partagés. En effet, le simulateur se limitait à l'indexation des données telle qu'elle est décrite dans le protocole Kademlia. Le travail réalisé sur ce point fut de créer, à la manière d'un véritable client P2P, pour chaque pair, une liste des blocs de données partagés.

#### 8.1.2 Modélisation des ressources et pollution

La modélisation des ressources au sein du simulateur PeerfactSim.KOM semble relativement réaliste. En effet, il est à noter que la taille des fichiers et la popularité sont incluses. Ce sont deux facteurs importants lorsque l'on s'intéresse aux réseaux P2P. La taille des fichiers influe, entre autre, sur le temps de transfert de ces derniers. De plus, on a vu que la pollution touche plus les fichiers populaires. On définit la popularité d'un fichier comme étant la probabilité qu'un utilisateur demande celui-ci. Les études à ce sujet montre

que la popularité semble suivre une loi de Zipf [26, 24]. Malheureusement, le simulateur n'inclut pas encore cette distribution. Cependant, il est possible d'utiliser une loi log-normale pour modéliser la popularité des données et se rapprocher de la réalité.

On dit qu'un fichier est pollué ou corrompu lorsqu'une partie ou la totalité de ce fichier diffère de l'original (censé, lui, être sain). On observe plusieurs types de pollutions :

- Les métadonnées (le titre ou les mots clés associés au contenu) ne correspondent pas au contenu,
- Le contenu est corrompu sans contenir de programme nuisible (typiquement lorsque le contenu est illisible en partie ou en totalité). On peut citer plusieurs types de corruption : la qualité est dégradée, le fichier est incomplet, du bruit a été inclus dans le fichier et le contenu a été mélangé. Lee *et al.* ont montré dans leur étude [26] que le taux de conscience est différent pour chaque type de dégradation.
- La ressource contient un programme nuisible adware, spyware, virus, vers, troyen, *etc.*

On a évoqué précédemment la problématique issue de la présence de pollution dans le système (hausse du trafic, baisse de la disponibilité de certaines ressources, trafic de contenu illégal, *etc.*).

Dans notre outil de simulation, la notion de corruption des données n'était pas prise en compte. Les ressources étaient modélisées par un couple  $\langle key, value \rangle$ , de la même manière que dans Kademia. Nous avons donc proposé une modélisation simplifiée de la pollution. Celle-ci reprend les deux premiers points évoqués ci-dessus, le troisième ne faisant pas partie de notre problématique. On considère le taux de pollution  $\tau$  présent dans un bloc de donnée. Celui-ci est représenté par un réel compris entre 0 et 1. On peut considérer ce taux comme étant le pourcentage de bits corrompus dans le fichier : 0 correspondant à un fichier tout à fait sain et 1 totalement pollué.

### 8.1.3 La procédure de téléchargement

La charge, en terme de trafic, des réseaux P2P provient pour une partie des messages protocolaires mais aussi des quantités de données qui sont échangées entre chaque utilisateur. L'indexation et l'échange de fichiers sont deux choses séparées dans le réseau Kad. L'indexation provient du protocole Kademia et le téléchargement, quant à lui, se fait directement entre deux pairs (voir section 1.4).

Le simulateur n'implantait pas de procédure de téléchargement, seul était codé le protocole d'indexation des données. De ce fait, et pour pouvoir mesurer l'impact, en terme de trafic, des pairs égoïstes et de la pollution, nous avons implanté une procédure de téléchargement. Celle-ci se décrit de la manière suivante :

- Une fois un bloc de donnée localisé (après la réponse de la procédure `find_value` du protocole Kademia), le pair envoie une requête à son homologue qui partage cette donnée.
- Lorsque ce dernier reçoit une requête, il vérifie qu'il partage toujours le block de donnée voulu et, si tel est le cas, l'envoie au pair demandeur.

On voit apparaître ici une première problématique au niveau de l'architecture de l'outil de simulation. En effet, il est fortement probable que le protocole d'indexation retourne plusieurs sources possibles pour le téléchargement d'un seul fichier et ce à des temps différents. Or voici ce qui se passe en réalité : le client P2P établit une liste des sources disponibles et l'utilisateur y fait son choix en fonction de différents critères comme par exemple la vitesse de téléchargement annoncée ou la place dans de la source dans la liste.

Hors, de la manière dont est implanté le simulateur - ceci provient du fait que c'est avant tout un simulateur P2P, voir précédemment - il manque ce niveau d'abstraction (le client P2P) entre le pair physique (l'ordinateur connecté au réseau) et l'utilisateur. Ainsi, pour le moment et du fait des contraintes de temps, l'utilisateur envoie une requête à chaque réponse du protocole d'indexation. On peut voir cela comme le pire des cas qui serait celui où chaque utilisateur demanderait un téléchargement d'un fichier à toutes les sources disponibles.

La seconde problématique issue du simulateur, surtout du fait de sa jeunesse, est l'absence de bande passante. La bande passante est la quantité d'information que peut véhiculer un canal par seconde, c'est le débit du réseau mesuré en bits par seconde. Dans la version dont nous disposons, celle-ci n'est pas encore implantée. C'est la bande passante qui détermine, avec la quantité de messages envoyés et le type de medium, la vitesse à laquelle se propagent les données. Cette idée de vitesse de propagation semble avoir une influence sur le comportement des utilisateurs ne serait-ce que dans le choix des sources par exemples.

Enfin, la dernière problématique concernant la procédure de téléchargement est plus de l'ordre de la modélisation. Comment faire lorsque le fichier demandé n'est plus partagé par l'hôte? Celui-ci doit-il, comme dans la réalité, ne rien faire quant à cette demande. Auquel cas, le pair demandeur doit avoir une certaine mémoire et une notion du temps pour pouvoir se rendre compte que sa demande a échoué. Ou, l'hôte envoie un message indiquant que la demande a échoué. Cette méthode, plus simple dans l'implantation ne risque-t-elle pas, au vu du nombre d'échanges effectués, d'introduire un biais dans la simulation? Pour le moment, du fait de l'architecture du simulateur et des contraintes de temps, c'est cette dernière méthode qui est utilisée.

## 8.2 Mesures

Dans ce genre de systèmes, où le nombre d'entité est élevé, il nous a semblé nécessaire d'éviter d'alourdir la simulation par un nombre conséquent de mesures. Ainsi, nous nous sommes focalisés sur les mesures existantes au sein du simulateur. La liste suivante introduit les mesures qui semblent intéressantes vis-à-vis de notre problème.

- Le nombre de pairs actifs (*i.e* connectés au réseau) par intervalle de temps. Ce paramètre est utile pour avoir une référence.
- Le nombre et le type de messages envoyés par intervalle de temps. Cela nous permet de faire la distinction entre charge issue du protocole d'indexation et de la procédure d'échange de données.
- Le nombre total de messages reçus par pair. Cela nous permet d'évaluer la répartition de la charge sur chaque utilisateur.
- Le ratio nombre de requêtes satisfaites sur le nombre total de requêtes émises. Ce critère permet d'évaluer l'efficacité du système.

## 8.3 Synthèse

Cette partie met en exergue les différentes propositions (au niveau de l'implantation) qui tentent de répondre aux diverses problématiques énoncées aux chapitres précédents. On pose aussi la question des mesures intéressantes et du compromis à faire entre le nombre d'entités présentes dans le système et de l'influence des mesures sur la simulation. Le chapitre suivant évoque les différentes expérimentations effectuées.

# Chapitre 9

## Expérimentations

### 9.1 Protocole expérimental

Cette partie décrit les paramètres qui peuvent influencer sur le système (voir la liste ci-après) et un scénario référent. Ce scénario est relativement simple, il comprend trois périodes principales : la première période est une phase dans laquelle un certain nombre d'agents se connecte au réseau. La deuxième concerne la publication de fichiers. Certains pairs mettent à disposition des fichiers. Ce sont les sources de téléchargement initiales. La troisième période concerne les téléchargements en eux-mêmes. Durant celle-ci, certains pairs demandent des fichiers, les téléchargent et les mettent à disposition (ou non). On espère ainsi englober toutes les phases du processus d'échange de données qui se déroule dans les systèmes P2P. Le tableau 9.1 montre ces trois périodes, leur durées et définit les quantités d'agents qui y participent. Les sections suivantes présentent les diverses expérimentations effectuées.

- Le nombre de pairs présents dans le système  $N$ .
- Le nombre de fichiers présents dans le système  $F$ ; sachant que la popularité des fichiers n'est pas uniforme et suit, pour le moment, une loi log-normale ( $m = 0,82$  et  $\sigma = 2,90$ )<sup>2</sup> sur l'intervalle  $[0, F]$ .
- Le taux d'empoisonnement  $E$  (*i.e* le pourcentage de fichiers pollués); sachant que le taux de pollution  $\tau$  d'un fichier corrompu est uniformément réparti sur l'intervalle  $[0, 1]$ .
- Le niveau de conscience de la pollution par agent  $\beta$ .
- Le nombre de pairs qui publient des données :  $P$ .
- Le nombre de fichiers publiés par chaque pair :  $fp$ .
- Le nombre de fichiers demandés par pair :  $r$  (*i.e* le nombre de requêtes envoyées par pair).
- Le nombre de pairs qui téléchargent mais ne partagent pas leurs données :  $Ego$ .
- La probabilité de redemander un fichier lorsque le pair s'est aperçu que celui-ci était pollué (et l'a donc supprimé).

### 9.2 Premières expérimentations

Dans cette première série d'expérimentations, on se propose de valider l'approche conceptuelle. On se pose la question suivante : est-ce que notre modélisation du problème et l'outil qui en découle semblent avoir un comportement réaliste ? Pour y répondre, on effectue plusieurs séries d'expérimentations. Dans chaque série on ne fera varier qu'un seul paramètre et on confrontera les résultats avec les résultats attendus.

---

<sup>2</sup>Valeurs issues des travaux Vasilios Darlagiannis sur le simulateur, voir [14] page 141

Étape	Durée (TS)	Action
e1	1h	$N$ pairs se connectent
e2	1min	$P$ pairs publient chacun $fp$ fichiers
e3	1h	$N/2$ pairs demandent chacun $r$ fichiers.
e3 (bis)	1h	$N/2$ pairs demandent chacun $r$ fichiers.

FIG. 9.1 – Scénario général

Ainsi, on se propose d’aborder la question de l’impact du nombre de pairs qui publient les données dans un système P2P coopératif (*i.e.* tout le monde partage) et sain (*i.e.* aucun fichier n’est pollué). L’intuition de départ est la suivante : en théorie, plus le nombre de pairs qui publient des ressources au départ est important, moins la charge par utilisateur est élevée. En effet, ceci peut s’expliquer par le fait que tout pair qui a téléchargé une ressource devient, dans ce cas, une nouvelle source. De cette manière la diffusion des données est plus rapide et la charge est censée se distribuer sur toutes les sources.

Pour cela, on reprend le scénario général (tableau 9.1) dans lequel on fixe tous les paramètres (voir tableau 9.2) sauf le nombre  $P$  que l’on fera varier de la manière suivante :  $P = N/100, N/50, N/20, N/10, N/2$  et enfin  $P = N$ .

Paramètre	Valeur
E	0
Ego	0
$\beta$	0
fp	1
r	1

FIG. 9.2 – Premières expérimentations. Paramètres fixes

On effectue donc plusieurs séries d’expérimentations pour cette première partie expérimentale. Le tableau 9.3 indique les valeurs des paramètres  $N$  et  $F$  associées à chaque série. Pour chaque série, on effectuera 5 simulations avec des graines aléatoires différentes.

Séries	Nombre de pairs $N$	Nombre de fichiers $F$
séries 1.1 à 1.3	100	10, 50 et 100
séries 2.1 à 2.3	500	50, 250 et 500
séries 3.1 à 3.3	1000	100, 500 et 1000
séries 4.1 à 4.3	5000	500, 2500 et 5000
séries 5.1 à 5.3	10000	1000, 5000 et 10000

FIG. 9.3 – Nombre de nœud du réseau P2P et nombre de fichiers disponibles pour chaque série d’expériences

### 9.3 Premiers résultats

Les premiers résultats semblent confirmer notre intuition de départ (répartition de la charge, voir précédemment). Cependant, les expérimentations étant toujours en cours, il ne paraît pas raisonnable

d'avancer, pour le moment, une quelconque relation entre les paramètres  $N$ ,  $P$  et  $F$ . Néanmoins, les premières expériences menées semblent corroborer l'idée que la démarche conceptuelle de modélisation des réseaux P2P via les SMA peut porter ces fruits.

## 9.4 Passage à l'échelle de l'outil

La problématique du passage à l'échelle, la capacité de celui-ci d'intégrer un nombre d'agents de plus en plus important vis-à-vis des performances, est relativement sensible. En effet, les tailles des réseaux P2P évoluant entre  $10^3$  et  $10^6$  pairs, la question de savoir si l'outil permet, en un temps et une consommation des ressources raisonnables, de simuler des réseaux de tailles réalistes est cruciale. Les premières expériences semblent montrer que le temps d'exécution des simulations varie linéairement, non pas avec le nombre de nœuds mis en jeu mais avec le nombre de messages échangés. En raison des contraintes de temps, et les simulations étant toujours en cours, ce point sera évoqué dans les perspectives.

# Conclusion et perspectives

## 1 Bilan

Le travail effectué au cours de ce stage s'est déroulé en trois grandes phases. Tout d'abord, la proposition d'une méthode de modélisation : après une présentation du domaine des réseaux P2P et des différentes problématiques qui s'y rapportent (coopération et pollution) nous avons exposé le besoin de modéliser le comportement des utilisateurs au sein des réseaux P2P. Nous avons examiné les études déjà réalisées dans ce domaine et en avons tiré un constat : il nous semble impératif d'intégrer dans notre modèle à la fois un comportement de l'utilisateur réaliste mais aussi une modélisation du réseau relativement correcte. Cependant, les modélisations des systèmes P2P existantes qui incluent ces deux niveaux d'abstraction sont quasi-inexistantes. Le premier point à donc consisté en une proposition d'une modélisation des systèmes P2P via une approche multi-agents. Dans cette partie, nous avons mis en exergue les problématiques issues de la complexité et des caractéristiques multi-échelles de ce type de systèmes.

La deuxième partie a consisté en la définition d'un cadre d'étude (coopération / pollution) et en la recherche d'un outil de simulation au travers duquel nous pourrions montrer la viabilité de notre approche conceptuelle et effectuer les premières simulations. Après avoir étudié les différents outils proposés dans les domaines des SMA et des réseaux P2P, il nous est apparu qu'aucun ne conviendrait en l'état à notre problématique. Nous avons donc fait le choix d'utiliser un simulateur P2P, PeerfactSim, que nous avons modifié pour pouvoir implanter notre modèle.

La troisième partie a donc consisté en la transformation d'un outil P2P en un simulateur plus spécifique à notre cadre d'étude. Dans cette partie, nous avons apporté une modélisation des notions de partage et de pollution, établi un modèle d'utilisateur basé sur un modèle d'agent réactif puis intégré ces dernières au simulateur. Nous avons ainsi pu poser les briques de base d'un simulateur intégrant les deux niveaux d'abstraction : comportement de l'utilisateur et réseau. Il nous est désormais possible d'étudier l'impact du comportement de l'utilisateur sur le fonctionnement des réseaux P2P. Enfin, nous avons établi un protocole expérimental qui, nous l'espérons, nous permettra, d'une part, de valider notre approche conceptuelle et d'autre part d'étudier la question de l'influence de la coopération et de la pollution sur le fonctionnement du réseau P2P. Ces expérimentations sont actuellement toujours en cours, mais les premiers résultats semblent encourageants et nous espérons pouvoir d'ici peu valider nos intuitions.

## 2 Limites

Cette approche étant relativement récente, il est à noter certaines limites concernant ce travail. La première limite concerne la modélisation en elle-même. En effet, celle présentée dans ce mémoire n'est qu'un premier pas effectué dans ce sens et de ce fait nécessite certainement discussion et approfondissement - notamment sur la prise en compte d'une couche d'abstraction liant les deux modèles réseau et comportement de l'utilisateur. La deuxième concerne l'outil en lui-même, au vu des contraintes de temps qui nous ont poussées à obtenir des résultats en terme d'expérimentation, l'outil est relativement centré sur notre cas d'étude (coopération et pollution). Ensuite, cet aspect rejoint la première limite évoquée

ci-dessus, l'architecture même de l'outil est à développer. Nous avons évoqué la problématique de l'interface entre le modèle de réseau et celui du comportement de l'utilisateur. Il semble opportun d'intégrer une nouvelle couche dans l'architecture du simulateur modélisant cette interface. La problématique qui s'en suit concerne le passage à l'échelle de l'outil : comment intégrer cette nouvelle couche sans nuire aux performances du simulateur ?

## 3 Perspectives

### 3.1 À court terme

Tout d'abord, on dispose à présent de toutes les briques nécessaires à la simulation des comportements égoïstes, d'une part, et de la présence de pollution. Ce stage prenant fin au mois d'août, il reste donc du temps pour étudier la question originale qui était de mesurer l'influence réciproque de ces deux paramètres. Ensuite, il semble opportun d'étudier plus en profondeur, le passage à l'échelle de l'outil ainsi modifié. En effet, il serait intéressant de pouvoir simuler un réseau de taille réaliste et de comparer les résultats obtenus avec ceux effectués sur une taille de réseau réduite. Enfin, il serait intéressant d'étudier l'effet de la localité géographique d'une part des pairs qui partagent et mettent à disposition des ressources, et d'autre part des pairs pollueurs.

### 3.2 À plus long terme

On peut envisager d'élargir notre approche au domaine de l'informatique ambiante et plus particulièrement aux réseaux ad-hoc et réseaux mesh. Les problématiques y sont différentes mais les besoins en terme de modélisation sont quasiment identiques. En effet, ces architectures nécessitent, en plus du comportement des utilisateurs, d'inclure la mobilité des appareils, et leurs comportements. On peut se poser la question au point de vue énergie : quel comportement adopter pour consommer le moins d'énergie possible ou encore : que se passe-t-il quand la réserve d'énergie devient insuffisante pour offrir les services voulus ?

# Table des figures

1.1	Architecture Hybride . . . . .	4
1.2	Architecture Décentralisée . . . . .	5
1.3	Classification résumée des systèmes P2P d'échange de fichiers . . . . .	6
1.4	Arbre binaire Kademia . . . . .	7
1.5	Lookup Kademia . . . . .	8
4.1	Système P2P via les SMA . . . . .	17
5.1	Caractéristiques techniques . . . . .	20
5.2	Architecture Peerfactsim . . . . .	21
9.1	Scénario général . . . . .	30
9.2	Premières expérimentations. Paramètres fixes . . . . .	30
9.3	Nombre de nœud du réseau P2P et nombre de fichiers disponibles pour chaque série d'expériences . . . . .	30

# Liste des abréviations utilisées

**ADSL** Asymmetric Digital Subscriber Line

**CAN** Content Addressable Network

**DHT** Distributed Hash Table

**ed2k** eDonkey2000

**IP** Internet Protocol

**SN** Super Nœud

**P2P** Peer-to-Peer

**PFS** Peerfactsim

**RPC** Remote Procedure Call

**SMA** Système Multi-Agents

**TS** Temps Simulé

# Bibliographie

- [1] <http://www.emule-project.net>.
- [2] <http://www.isi.edu/nsnam/ns/>.
- [3] <http://www.massivesoftware.com/>.
- [4] The madkit project <http://www.madkit.org/>.
- [5] PeerfactSim.KOM : A Simulator for Large-Scale Peer-to-Peer Networks. <http://www.peerfactsim.com>.
- [6] Repast agent simulation toolkit <http://repast.sourceforge.net/>.
- [7] <http://peersim.sourceforge.net/>, 2007.
- [8] E. Adar and B. Huberman. Free riding on gnutella, 2000.
- [9] Stephanos Androutsellis-Theotokis and Diomidis Spinellis. A survey of peer-to-peer content distribution technologies. *ACM Comput. Surv.*, 36(4) :335–371, 2004.
- [10] N. Ben Azzouna and F. Guillemin. Experimental analysis of peer-to-peer applications on traffic in commercial ip networks. Technical report, France Telecom, France Telecom, Division R&D, 2 avenue Pierre Marzin F-22300 Lannion, France., 2004.
- [11] Ozalp Babaoglu, Hein Meling, and Alberto Montresor. Anthill : A framework for the development of agent-based peer-to-peer systems. In *ICDCS '02 : Proceedings of the 22 nd International Conference on Distributed Computing Systems (ICDCS'02)*, page 15, Washington, DC, USA, 2002. IEEE Computer Society.
- [12] Monique Becker and André-Luc Beylot. *Simulation des Réseaux*. Réseaux et Télécoms. Hermès, <http://www.editions-hermes.fr>, février 2006. Traité IC2.
- [13] J. Chuang. In search of homo swappus : Evolution of cooperation in peer-to-peer systems. In *IEEE Conference on P2P Computing*, september 2005.
- [14] Vasilios Darlagiannis. *Overlay Network Mechanisms for Peer-to-Peer Systems*. PhD thesis, Vom Fachbereich Informatik der Technischen Universität Darmstadt, 2005.
- [15] Y. Demazeau. From interactions to collective behaviour in agent-based systems, 1995.
- [16] Gireg Desmeulles. *Réification de interactions pour l'expérience in virtuo de systèmes biologiques multi-modèles*. PhD thesis, Université de Bretagne Occidentale, 2006.
- [17] Van Dyke, Robert Savit, and Rick L. Riolo. Agent-based modeling vs. equation-based modeling : A case study and users' guide. In *MABS*, pages 10–25, 1998.
- [18] Michal Feldman and John Chuang. Overcoming free-riding behavior in peer-to-peer systems. *SIGecom Exch.*, 5(4) :41–50, 2005.
- [19] Michal Feldman, Christos Papadimitriou, John Chuang, and Ion Stoica. Free-riding and whitewashing in peer-to-peer systems. In *PINS '04 : Proceedings of the ACM SIGCOMM workshop on Practice and theory of incentives in networked systems*, pages 228–236, New York, NY, USA, 2004. ACM Press.
- [20] Jacques Ferber. Les systèmes multi-agents : un aperçu général. *Techniques et sciences informatiques*, 16(8) :979–1012, 1997.

- [21] D. Gaïti and L. Merghem-Boulahia. *L'autonomie dans les réseaux*, chapter 5, pages 133–165. Hermes Lavoisier, 2006.
- [22] Thomas Gschwind and Cecilia Mascolo, editors. *Software Engineering and Middleware, 4th International Workshop, SEM 2004, Linz, Austria, September 20-21, 2004, Revised Selected Papers*, volume 3437 of *Lecture Notes in Computer Science*. Springer, 2005.
- [23] David Hales. From selfish nodes to cooperative networks " emergent link-based incentives in peer-to-peer networks. In *P2P '04 : Proceedings of the Fourth International Conference on Peer-to-Peer Computing (P2P'04)*, pages 151–158, Washington, DC, USA, 2004. IEEE Computer Society.
- [24] Daniel Hughes, Geoff Coulson, and James Walkerdine. Free riding on gnutella revisited : The bell tolls? *IEEE Distributed Systems Online*, 6(6) :1, 2005.
- [25] Odile Lausecker Laurent Viennot. Les réseaux pair-à-pair, <http://interstices.info/p2p>.
- [26] Uichin Lee, Min Choiz, Junghoo Choy, M. Y. Sanadidiy, and Mario Gerla. Understanding pollution dynamics in p2p file sharing. In *5th International Workshop on Peer-to-Peer Systems (IPTPS'06)*, Santa Babara, CA, USA, February 2006.
- [27] Jian Liang. Pollution in p2p file sharing systems, 2005.
- [28] M. Luck, P. McBurney, O. Shehory, and S. Willmott. *Agent Technology : Computing as Interaction (A Roadmap for Agent Based Computing)*. AgentLink, 2005.
- [29] P. Maymounkov and D. Mazieres. Kademia : A peer-to-peer information system based on the xor metric, 2002.
- [30] Gianluca Moro and Manolis Koubarakis, editors. *Agents and Peer-to-Peer Computing, First International Workshop, AP2PC 2002, Bologna, Italy, July, 2002, Revised and Invited Papers*, volume 2530 of *Lecture Notes in Computer Science*. Springer, 2003.
- [31] S. Naicken, A. Basu, B. Livingston, and S. Rodhetbhai. A survey of peer-to-peer network simulators. *Proceedings of The Seventh Annual Postgraduate Symposium, Liverpool, UK*, 2006.
- [32] S. Naicken, A. Basu, B. Livingston, S. Rodhetbhai, and I. Wakeman. Towards yet another peer-to-peer simulator. *Proceedings of The Fourth International Working Conference on Performance Modelling and Evaluation of Heterogeneous Networks (HET-NETs), Ilkley, UK*, 2006.
- [33] Pierre GLIZE Olivier BOISSIER, Sylvain GITTON. *Caractéristiques des systèmes et des applications*, volume 29 of *Arago*, pages 25 – 48. Observatoire Français des Technologies Avancées, 2004. isbn : 978-2906028159.
- [34] L. Ramaswamy and L. Liu. Free riding : A new challenge to peer-to-peer file sharing systems, 2003.
- [35] Jean Pierre Sansonnet. <http://www.limsi.fr/jps/>, 2005.
- [36] S. Saroiu, P. Gummadi, and S. Gribble. A measurement study of peer-to-peer file sharing systems, 2002.
- [37] Mario Schlosser and Sepandar Kamvar. Simulating a file-sharing p2p network.
- [38] Evan Sultanik, Maxim Peysakhov, and William C. Regli. Agent transport simulation for dynamic peer-to-peer networks. In Jaime Simão Sichman and Luis Antunes, editors, *MABS*, volume 3891 of *Lecture Notes in Computer Science*, pages 162–173. Springer, 2005.
- [39] N. San Ting. Simulating peer-to-peer networks. Master's thesis, Department of Computer Science University of Saskatchewan Saskatoon, Saskatchewan Canada., 2006.