Université Abou Bekr Belkaid
Tlemcen Algérie
جامعة أبي بكر بلقايد
تلمسان الجزائر

République Algérienne Démocratique et Populaire

Université Abou Bakr Belkaid– Tlemcen

Faculté desSciences

Département d'Informatique

Mémoire de fin d'études

Pour l'obtention du diplôme de Master en Informatique

Option: Réseaux et Systèmes Distribués (R.S.D)

Thème

Partage de fichiers de pair à pair sur JXTA

Réalisépar :

Muhamadu Aly Jalo

Présenté le 22 Juin 2016 devant le jury composé de:

- *Mme Didi Fedoua* — *Président*
- *Mr Belhoucine Amine* — *Encadrant*
- *Mr Benamar Abdelkrim* — *Examinateur*
- *Mme Malti Djawida* — *Examinateur*

Année universitaire: 2015-2016

Remerciements

Premièrement je tiens à remercier Dieu le tout puissant et miséricordieux pour le courage, la force, la foi et la volonté pour accomplir ce travail.
Je remercie aussi à tous mes enseignants qui m'ont transmis des connaissances depuis l'année d'apprentissage de la langue française jusqu'à master 2, je tiens a remercié très chaleureusement mon tuteur Monsieur Belhocine pour tous les conseils et orientations pendants ce travail. Son exigence m'a grandement stimulée. Et aussi je tiens à remercie les membres des jurys qu'ont accepté d'évaluer ce travail c'est un grand honneur pour moi.
Je remercie aussi ma mère et mon père leur prière et leurs bénédictions m'ont été d'un grand secours pour mener à bien mes études.
Je remercie mes amis Imine Youcef et Mohamed Kaddour, pour leur soutien moral, leurs opinions, leurs aides et la force et le courage qu'ils m'ont donnée pendants ce travail.

Dédicaces

Je dédie ce travail à mes estimables parents dont aucune dédicace ne saurait exprimer l'amour, l'estime, le dévouement et le respect que j'ai toujours eu pour vous.

Rien au monde ne vaut les efforts fournis jour et nuit pour mon éducation et mon bien être.

Ce travail est le fruit de vos sacrifices que vous avez consentis pour mon éducation et ma formation.

Mais aussi à mes frères et sœurs spécialement à Aminata Jalo en témoignage de l'attachement, de l'amour et l'affection que je porte pour toi. Malgré la mort qui ta séparé de moi au début de cette année universitaire, vous êtes toujours dans mon cœur. Ainsi que toute ma famille.

À tous ceux qui de près ou de loin m'ont soutenu de façon directe ou indirecte.

À la promotion RSD 2015/2016, à toutes les communautés bissau-guinéennes, et aussi toutes les communautés étrangères du Tlemcen, à mes professeurs, et tous mes amis :

kayzer, ciro, josemar, joseph, homena, dilma, jado, ramex, ety, kankas, alamara, ensa, imine, kaddour, amine, hervé, demba, lo, lué, yanick, heldis, William, Nerissa, Rainatu, Code.

TABLE DES MATIERE

Table des figures

Liste des tableaux

Introduction générale

L'Internet est constitué des milliers de réseaux informatiques dans le monde entier qui sont Interconnectés les uns aux autres. Son utilisation augmente avec la population de l'humanité, alors cette augmentation cause des anomalies dans la disponibilité de l'espace de stockage des disques durs dans le cas où l'on utilise la technologie client/serveur. Imaginons que de nos jours l'on utilise que des serveurs pour partager nos ressources, alors on aura de sérieux problèmes comme par exemple des problèmes de stockages : combien de millions des données seront stocké dans le serveur par jour? Nos serveur ont des capacités des stockages limitées ça veut dire tôt ou tard nous allons arriver au bout de sa capacité à moins que à chaque fois on augmente sa capacité mais jusqu'à quand peut-on continuer à augmenter ? Et ça sera très couteux et fastidieux à maintenir. De plus, nous resteront toujours dépendants des serveurs alors qu'une fois qu'ils tombent en panne on a plus de fonctionnalité dans ce modèle sans oublier que le serveur sera toujours la cible donc il est très vulnérable.

Ceci a conduit à l'apparition du Peer-to-Peer (aussi connu sous le P2P) venue pour aider à surmonter des problèmes tels que le manque de bande passante et le coût de Logiciel / matériel sur Internet.

L'aspect le plus reconnu de P2P est ses applications de partage de fichiers:

Napster, eDonkey, eMule (basé sur eDonkey réseau) et LimeWire sont quelques exemples. P2P est l'informatique distributive qui peut être utilisé pour une variété de buts tels que le partage de fichiers, réduisant ainsi considérablement le coût.

Ce projet est basé sur cette nouvelle technologie.

Les connaissances acquises tout au long de la phase de recherche de ce projet ont aidé à concevoir, mettre en œuvre et développer une application unique et pleinement opérationnelle qui réalise le partage de fichiers P2P.

L'application développée dans ce projet est décentralisé (pas besoin d'un serveur principal). Cette application fait la découverte des pairs connectés, mais aussi le partage et le

téléchargement des ressources une fois que les pairs sont connectés avec succès. Pour atteindre cet objectif, ce projet utilisera la technologie JXTA, développé par Sun Microsystems. Cette technologie est appropriée pour développer une application basée sur le P2P. Ainsi, elle permettra aux appareils de partager des ressources sans connaissance de leur architecture réseau.

1.Chapitre 1 : Généralités

1.1. Introduction

Les progrès technologiques des ordinateurs et le haut débit permettent aux différentes applications sur des ordinateurs distincts (et distants) de coopérer pour effectuer des tâches coordonnées.

Avant les années 80, les ordinateurs étaient encombrants et chers (les systèmes centralisés)

A partir de la mi-80, deux nouveautés:

– Microprocesseurs (moins chers et très rapide)

– LAN et WAN

Les ordinateurs en réseaux sont non seulement faisables, mais simples. Alors dans ce chapitre nous allons parler des Systèmes distribués, leurs objectifs avantages et inconvénients mais aussi de leur classification où nous allons décrire les client-serveur et p2p leur avantages et inconvénients etc.

1.2. Définition

C'est quoi les Systèmes Distribués ?

Pour mieux comprendre le vrai sens de la définition du système distribué (SD) il faut bien savoir c'est quoi un système centralisé car le système distribué est l'opposé d'un système centralisé.

Alors parlons du système centralisé.

Un système est dit centralisé quand tout est localisé sur la même machine et accessible par le programme qui est un système logiciel s'exécutant sur une seule machine accédant localement aux ressources nécessaires (données, code, périphériques, mémoire ...)

Alors revenons sur notre sujet c'est quoi System Distribué (SD) ?

Une définition parmi les autres ?

Un Système Distribué (SD) est un ensemble d'ordinateurs indépendants connectés en réseau et communiquant via ce réseau. Cet ensemble apparaît du point de vue de l'utilisateur comme une unique entité. [1][2]

Vision matérielle d'un système distribué : architecture matérielle

Machine multiprocesseurs avec mémoire partagée.
Cluster d'ordinateurs dédiés au calcul/traitement massif parallèle.
Ordinateurs standards connectés en réseau.

Vision logicielle d'un système distribué
Système logiciel composé de plusieurs entités s'exécutant indépendamment et en parallèle sur un ensemble d'ordinateurs connectés en réseau.

Pourquoi des systèmes répartis ?

Aspects économiques
Adaptation de la structure d'un système à celle des applications
Besoin d'intégration
Besoin de communication et de partage d'information
Réalisation de systèmes à haute disponibilité
Partage de ressources (programmes, données, services)
Réalisation de systèmes à grande capacité d'évolution

Ce qu'offre un système réparti ?

- Transparence à la localisation : L'utilisateur ignore la situation géographique des ressources. Transparence à la migration.
- Transparence d'accès : L'utilisateur accède à une ressource locale ou distante d'une façon identique.

- Transparence à l'hétérogénéité : L'utilisateur n'a pas à se soucier des différences matérielles ou logicielles des ressources qu'il utilise, Notion d'interopérabilité.
- Transparence aux pannes (réseaux, machines, logiciels) : Les pannes sont cachées à l'utilisateur.
- Transparence à l'extension des ressources: Extension ou réduction du système sans occasionner de gêne pour l'utilisateur (sauf performance). [3]

1.3. OBJECTIFS

Coût: plusieurs processeurs à bas prix

Puissance de calcul: profite des ressources des ordinateurs.

Performance: calcul parallèle

Fiabilité: résistance aux pannes logicielles ou matérielles

Extensibilité: croissance progressive selon le besoin

1.4. Avantages [3]

- partage de données
- partage de périphériques
- communication
- souplesse (politiques de placements)

1.5. Inconvénients [3]

- réseaux : la saturation et délais mais aussi si le réseau est en panne alors le system plante.
- sécurité : piratage

1.6. Domaine d'application

Quelques domaines d'application des systèmes répartis [3][4]

– Coopération d'équipes pour la conception d'un produit. Exemple : CFAO (Conception et Fabrication Assistées par Ordinateurs)

– Production coopérative de documents

– Partage cohérent d’information

Gestion intégrée des informations d'une entreprise
– Intégration de l’existant
Contrôle et organisation d’activités en temps réel
Centres de documentation, bibliothèques
– Recherche, navigation, visualisation multimédia
Systèmes d’aide à la formation

1.7. Architecture Client-serveur et pair-a-pair (Peer to Peer ou p2p)

1.7.1.Client-serveur

L'environnement client-serveur désigne un mode de communication à travers un réseau entre plusieurs programmes ou logiciels : l'un, qualifié de client, envoie des requêtes ; l'autre ou les autres, qualifiés de serveurs, attendent les requêtes des clients et y répondent. Par extension, le client désigne également l'ordinateur sur lequel est exécuté le logiciel client, et le serveur, l'ordinateur sur lequel est exécuté le logiciel serveur.

En général, les serveurs sont des ordinateurs dédiés au logiciel serveur qu'ils abritent, et dotés de capacités supérieures à celles des ordinateurs personnels en ce qui concerne la puissance de calcul, les entrées-sorties et les connexions réseau. Les clients sont souvent des ordinateurs personnels ou des appareils individuels (téléphone, tablette), mais pas systématiquement. Un serveur peut répondre aux requêtes d'un grand nombre de clients.

Il existe une grande variété de logiciels serveurs et de logiciels clients en fonction des besoins à servir : un serveur web publie des pages web demandées par des navigateurs web ; un serveur de messagerie électronique envoie des mails à des clients de messagerie ; un serveur de fichiers permet de stocker et consulter des fichiers sur le réseau ; un serveur de données à communiquer des données stockées dans une base de données, etc.[5]

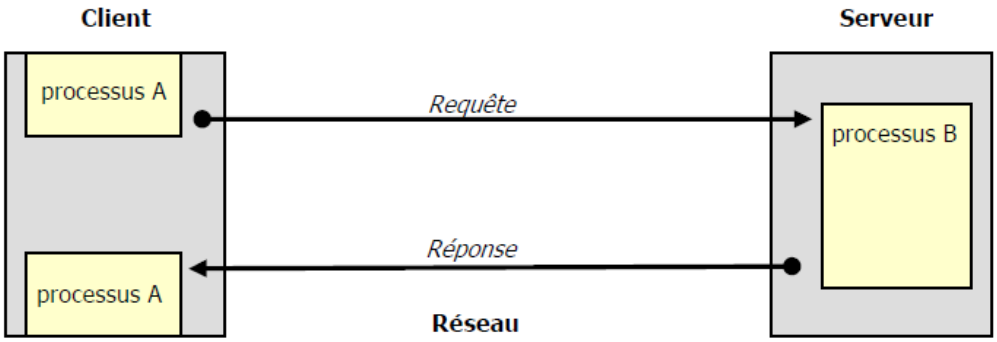


Figure 1.1: Fonctionnement

1.7.2. Caractéristiques des systèmes client serveur

Les éléments qui caractérisent une architecture client serveur sont [5][6]:

- **Service**

Ici on a une relation entre le logiciel sur des machines séparées connecté aux réseaux où les clients consomment ou utilisent les services qui sont fournie par les serveurs.

- **Partage de ressources**

Le serveur partage ces ressources ou les ressources des clients et il traite plusieurs requêtes des clients et contrôle leurs accès aux ressources

- **Protocole asymétrique**

Les serveurs sont en attente des requêtes des clients alors les clients doivent déclencher le dialogue ça veut dire demander les services ou bien envoyer des requêtes au serveur donc le protocole utilisé est asymétrique.

- **Transparence de la localisation**

L'architecture client-serveur doit masquer au client la localisation du serveur (que le service soit sur la même machine ou accessible par le réseau). Transparence par rapport aux systèmes d'exploitation et aux plates-formes matérielles. Idéalement, le logiciel client serveur doit être indépendant de ces deux éléments.

- **Message**

Client et serveur ont les messages comme moyen d'échanges entre eux.

❖ **Evolution**

L'évolution est parmi les plus essentiels des caractéristiques de l'architecture client serveur. Il est important d'avoir une évolution du nombre de clients mais aussi celle du nombre et des caractéristiques des serveurs.

1.7.3. Caractéristiques d'un processus serveur [6]:

- Le serveur est toujours en attente d'une connexion entrante sur un ou plusieurs ports. Ce(s) port(s) est (sont) en écoute de tentative des connexions lancé par les clients
- à la connexion d'un client sur le port en écoute, il ouvre un socket local au système d'exploitation;
- à la suite de la connexion, le processus serveur communique avec le client suivant le protocole prévu par la couche application du modèle OSI.

1.7.4.Caractéristiques d'un processus client [6]:

- il établit la connexion au serveur à destination d'un ou plusieurs ports réseaux qui sont en écoute.
- lorsque la connexion est acceptée par le serveur, le client communique via le protocole prévu par la couche applicative du modèle OSI.

Le protocole de communication qui doit être utilisé entre le client et le serveur au niveau de la couche transport du modèle OSI doit être le même. Un serveur est généralement capable de servir plusieurs clients simultanément. On parle souvent d'un service pour désigner la fonctionnalité offerte par un processus serveur. On définit aussi comme serveur, un ordinateur spécialisé ou une machine virtuelle ayant pour unique tâche l'exécution d'un ou plusieurs processus serveur.

1.7.5.Les différentes modèles du client-serveur

En fait les différences sont essentiellement liées aux services qui sont assurés par les serveurs.
On distingue couramment [6]:

- Client-serveur de donné:

Dans ces cas les serveurs assure des taches des gestions, stockage et de traitement des donnes. C'est les cas le plus connu des client-serveur et qui est connue par tous les grands SGBD : La base de donnes avec tous ses outils (maintenance, sauvegarde…) est installée sur un poste serveur. Sur les clients, un logiciel d'accès est installé est permettant d'accéder à la base des donnes du serveur. Tous les traitements sur les données sont effectués sur le serveur qui renvoie les informations demandées (souvent à travers une requête SQL) par les clients.

-Client-serveur de présentation :

Dans ce cas les présentations des pages affichées par les clients sont intégralement prises en charge par le serveur.

-Client-serveur de traitement :

Dans ce cas, les serveurs effectuent des traitements à la demande du client, il peut s'agir des traitements particuliers sur des données, des vérifications des formulaires de saisie etc. Ces traitements peuvent être réalisé par des programmes installée sur des serveurs ou bien intégrés dans les bases de donnée (exemple : triggers, procédure), mais dans ce cas la partie donné et traitement sont intégrée.

1.7.6.Avantages de l'architecture client-serveur

- Toutes les données sont centralisées sur un seul serveur, physique ou virtuel, ce qui simplifie les contrôles de sécurité, l'administration, la mise à jour des données et des logiciels.
- La complexité du traitement et la puissance de calculs sont à la charge du ou des serveurs, les utilisateurs utilisant simplement un client léger sur un ordinateur terminal qui peut être simplifié au maximum.
- Recherche d'information : les serveurs étant centralisés, cette architecture est particulièrement adaptée et véloce pour retrouver et comparer de vaste quantité d'informations (moteur de recherche sur le Web), ce qui semble être un inconvénients pour le P2P beaucoup plus lent, à l'image de Freenet.

1.7.7.Inconvénients de l'architecture client-serveur [7]

- Ici on a le problème de dynamisme car si trop de clients veulent communiquer avec le serveur au même moment, ce dernier risque de ne pas supporter la charge (alors que les réseaux pair-à-pair fonctionnent mieux en ajoutant de nouveaux participants).
- Mais aussi le problème de panne car ici on a le serveur alors il est indispensable qu'il soit actif à tout moment, au cas contraire par exemple si le serveur n'est plus disponible, plus aucun des clients ne fonctionne (le réseau pair-à-pair continue à fonctionner, même si plusieurs participants quittent le réseau).
- Les coûts de mise en place et de maintenance peuvent être élevés.
- Pas des communications directes entre les clients. En aucun cas, les clients ne peuvent communiquer directement entre eux et doivent toujours passer par le serveur, ce qui entraine une asymétrie de l'information au profit des serveurs.

1.7.8.Exemple :

- La navigation sur internet, en occurrence la consultation de pages sur un site web fonctionne sur une architecture client-serveur. Un internaute connecté au réseau facebook avec son ordinateur représente le client, le serveur est constitué par le ou les ordinateurs contenant les applications qui délivrent les pages demandées. Dans ce cas, c'est le protocole de communication HTTP ou XML socket qui est utilisé.

Figure 1.2: Architecture client-serveur

1.8. Pair-a-pair (peer to peer p2p)

1.8.1.Définition

Dans les recherches réalisées pour entamer ce mémoire, différentes approches ont été trouvées pour définir un réseau P2P. Voici deux définitions parmi beaucoup d'autres [8][9]:
" Le terme "Peer-to-Peer", "poste à poste" ou "pair à pair" en français (ou plus couramment P2P), désigne un modèle de réseau informatique dont les éléments (les nœuds ou "peer") sont à la fois clients et serveurs lors des échanges. Grâce à ce modèle on a pu décentraliser les réseaux, en opposition aux architectures traditionnelles client/serveur. Ce modèle a pu nous offrir la possibilité des pairs avoir des communications directes, d'égal à égal comme son nom indique, entre les différents nœuds du réseau, qui peuvent alors échanger différents types d'informations sans passer par un serveur central."

"Peer-to-peer systems are distributed systems consisting of interconnected nodes able to self-organize into network topologies with the purpose of sharing resources such as content, CPU cycles, storage and bandwidth, capable of adapting to failures and accommodating transient populations of nodes while maintaining acceptable connectivity and performance, without requiring the intermediation or support of a global centralized server or authority".
L'utilisation des réseaux Peer-to-Peer est aujourd'hui en pleine croissance à cause de leurs avantages et caractéristiques et leur propagation dans tous les domaines. Pour cela plusieurs logiciels ont été développés.

1.8.2. Principe général

Des logiciels particuliers sont nécessaires lors de l'utilisation d'un système pair-à-pair pour les nœuds qui font partie de ce système. Ces logiciels sont très importants pour avoir le système p2p. Ces logiciels, créent ce qu'on appelle dans ce domaine du p2p le servent (de la contraction de « *serv*eur » et de « cli*ent* »), alors il permet à la fois les fonctions de *client* et de *serveur*. L'un des plus grands avantage de ce système comme son nom l'indique pair-à-pair est: les communications et les échanges qui se font entre des nœuds qui ont la même responsabilité dans le système.

En particulier, les systèmes de partage de fichiers pair-à-pair permettent de rendre les objets d'autant plus disponibles qu'ils sont populaires, et donc répliqués sur un grand nombre de nœuds. Cela permet alors de diminuer la charge (en nombre de requêtes) imposée aux nœuds partageant les fichiers populaires, ce qui facilite l'augmentation du nombre de nœuds et donc de fichiers dans le réseau. C'est ce qu'on appelle le passage à l'échelle.

Le modèle pair-à-pair va bien plus loin que les applications de partage de fichiers : il permet en effet de décentraliser des services et de mettre à disposition des ressources dans un réseau, nommées *objets*. Tout nœud d'un réseau pair-à-pair peut alors proposer des objets et en obtenir sur le réseau.

Les systèmes pair-à-pair permettent donc de faciliter le partage d'informations. Ils rendent aussi la censure ou les attaques légales ou pirates plus difficiles. Ces atouts font des

systèmes pair-à-pair des outils de choix pour décentraliser des services qui doivent assurer une haute disponibilité tout en permettant de faibles coûts d'entretien. Toutefois, ces systèmes sont plus complexes à concevoir que les systèmes client-serveur. [10]

1.8.3.Les avantages du P2P :

-Les communications sont directes

-Décentralisation

-Passage à l'échelle

-La réplication, redondance des données

-Un nœud peut accéder directement à un ou plusieurs nœuds.

-Si une machine tombe en panne, cela ne remet pas en cause l'ensemble du système.

-Possibilité de créer des groups

1.8.4.Les Inconvénients du P2P

-Pas de Qualité du Service (QoS).

-problèmes de sécurité

-Les temps de localisation sont plus longs

1.8.5.Caractéristique de réseau P2P

La popularité du système pair à pair (P2P) ne cesse d'augmenter grâces aux nombreux avantages et caractéristiques de ce dernier. On trouve les caractéristiques suivantes [10] [11]:

- Grande échelle : des grands nombre des nœuds coopèrent pour partager des ressources avec une bonne performance du system. Cela signifie qu'un système P2P doit offrir des méthodes bien adaptées avec un environnement dans lequel il y a un grand volume de données à partager, un nombre important de messages à échanger entre un grand nombre de nœuds partageant leurs ressources via un réseau largement distribué.

- Autonomie de nœuds : l'autonomie des nœuds permet aux nœuds d'être indépendants dans la gestion de leurs ressources d'une façon libre. Il décide quelle partie de ses données à partager. Il peut se connecter ou/et se déconnecter à n'importe quel moment.

- Environnement dynamique : à cause de l'autonomie de nœuds, chaque nœud peut quitter le système à n'importe quel moment ce qui fait disparaitre ses ressources du système. De

nouvelles ressources peuvent être ajoutées au système lors de la connexion de nouveaux nœuds. Alors, à cause de l'instabilité des nœuds, les systèmes P2P doivent être capables de gérer un grand nombre de ressources fortement variables. La sortie d'un nœud du système (ou la panne d'un nœud) ne doit pas mettre le système en échec. Elle doit être tolérée et avoir un "petit" impact sur la performance de tout le système.

- Hétérogénéité: à cause de l'autonomie des nœuds possédant des architectures matérielles et/ou logicielles hétérogènes, les systèmes P2P doivent posséder des techniques convenables pour résoudre les problèmes lies à l'hétérogénéité de ressources.

- Décentralisation : le fait que chaque nœud gère ses propres ressources permet d'éviter la centralisation du contrôle. Un système P2P peut fonctionner sans avoir aucun besoin d'une administration centralisée ce qui permet d'éviter les goulots d'étranglements et d'augmenter la résistance du système face aux pannes et aux défaillances.

- Auto configuration : puisque les systèmes P2P sont souvent déployés sur l'Internet, la participation d'un nouveau nœud à un système P2P ne nécessite pas une infrastructure couteuse.

Il suffit d'avoir un point d'accès à l'Internet et de connaitre un autre nœud déjà connecte pour se connecter au système. Un système P2P doit être un environnement ouvert ; c'est-à-dire, un utilisateur sur un nœud doit être capable de connecter son nœud au système sans avoir besoin de contacter une personne et sans avoir besoin de passer par une administration centralisée.

Figure 1.3: Illustration du modèle P2P

1.9. Comparaison entre les réseaux P2P et l'architecteur Client/serveur

Modèle p2p	Modèle Client/serveur
Architecture décentralisée	Architecture centralisé
Topologie Dynamique	Topologie statique
Auto-organisé	Supervisé
Les nœuds sont indépendants	Les nœuds dépendent des serveurs
Les paires partagent et consomment des ressources	Les clients ne partage jamais les ressources
Attaques Difficiles	Attaque faciles

Tableau 1.1 : Comparaison entre P2P et Client/serveur

1.10. Conclusion

Le Système Distribué est le grand succès des nos jours dans les services des réseaux informatique grâce a son arrivé nous avons réussi a décentralisé le système informatique et augmenter les capacités dans les performances des calculs et de traitements des ordinateurs. Dans ce chapitre nous avons étudié les objectifs, les domaines d'applications des systèmes distribués, nous avons aussi décrit le modèle client/serveur où nous avons parlé de ses caractéristiques, ses avantages où il a été remarqué que la faiblesse de ce modèle est la centralisation, de ce fait il a un grand problème car il est vulnérable à une attaque qui peut rendre le serveur inaccessible et une fois que le serveur est en panne c'est tout les system que est indisponible. Enfin,

nous avons parlé du modèle p2p qui est un modèle qui nous intéresse et sur lequel est basé ce travail. Ce modèle permet la communication directe entre les clients et nous permet ainsi de nous passer des services offerts par les serveurs.

2. Chapitre 2: Les Réseaux P2P

2.1. Introduction

Avec l'arrivée de l'Internet et son rôle actif croissant dans l'économie et la société, les réseaux informatiques n'ont eu de cesse de trouver des innovations pour exploiter les ressources qu'un réseau de cette ampleur contient.

Les réseaux pair-à-pair ont beaucoup évolué ces dernières années à tel point que les dernières générations de systèmes pair-à-pair fournissent à leurs usagers des applications de « télévision sur Internet » et de « Vidéo-à-la-demande ».

Que ce soit pour un usage grand public tel que les deux exemples cités précédemment ou bien pour des applications scientifiques à fin de répartir des calculs ou stocker des données, il est de nos jours de plus en plus courant d'avoir la nécessité de recourir à l'utilisation de réseaux pair-à-pair.

L'objectif de cette étude est de faire comprendre le phénomène du peer-to-peer et faire en sorte que les futures étudiantes comprennent mieux cette notion de peer-to-peer.

Nous allons étudier dans ce chapitre l'arrivée du peer-to-peer dans le contexte de l'Internet en rappelant brièvement son historique, sa définition ainsi que les différents types de réseaux peer-to-peer où nous pourrons observer les différents inconvénients et avantages de ces différents types au sein réseaux peer-to-peer.

2.2. Historique des réseaux pair-à-pair

Le P2P est devenu très populaire avec Napster crée par ShwanFanning, en 1999. Logiciel pour le partage de musique en ligne, est le plus grand taux de croissance de tous les temps. Les téléchargements des musiques est sur 24h/24 et 7j/7. Napster a eu des poursuites judiciaires suite le problème de copyright alors Napster devait payer 26 millions de dollars aux compositeurs et éditeurs en 2001, et en 2002 la fermeture de Napster.

Après les problèmes rencontrés par Napster, émergence de nombreux autres programmes P2P, parmi lesquels Gnutella et Kazaa.

Ces applications utilisant une approche plus décentralisée, permettent de rendre un contrôle de police plus difficiles.

Début 2000 : Naissance de Kazaa.

Offre la possibilité nouvelle de pouvoir télécharger un même fichier depuis sur plusieurs sources afin d'accroître la vitesse de réception.

2003 : Apparition d'eMule et d'eDonkey : nouvelle façon de télécharger les fichiers. Il s'agit de la technique de "fractionnement de fichier" : à peine un téléchargement est commencé que la partie récupérée est disponible pour l'envoi. Cette méthode permet d'augmenter encore la vitesse de transfert de fichiers. [10]

2.3. Les applications Peer-to-Peer

Plusieurs domaines d'applications utilisent le principe de réseaux P2P, il y a par exemple :

- Le partage de fichiers avec des logiciels comme KaZaA ou eMule
- Les logiciels de téléphonie sur Internet dont le plus populaire Skype.
- Des programmes de messageries instantanées (ICQ, AIM)
- La Télévision par P2P: Les fondateurs de Kazaa et de Skype préparent la télévision via le peer-to-peer ("Vénice Project")
- Des moteurs de recherche tels qu'Amoweba (moteur de recherche distribué en P2P basé sur l'utilisation intelligente des liens favoris des internautes).
- Des projets permettant le partage de la puissance de calcul d'ordinateur du monde entier, tels que le "projet SETI" dont le but est de recherche des traces d'Intelligence Extraterrestre.
- Des Plate-forme de développement tel que "JXTA3", technologie développée par Sun Microsystems, qui a pour but de pouvoir interconnecter n'importe quel système sur n'importe quel réseau. Elle utilise des protocoles basés sur XML et doit permettre de mettre en relation des ordinateurs, des téléphones portables, des PDA, etc. pour les faire communiquer de manière décentralisée. "Java Binding" est actuellement la version la plus aboutie. . [11]

2.4. Classification de l'architecture P2P:

Nous pouvons classer les systèmes informatiques dans (2) deux catégories distinct : systèmes centralisés et systèmes distribués, à leur tour les distribués, sont divisés en deux: le modèle client/serveur et le modèle pair à pair [11][12].

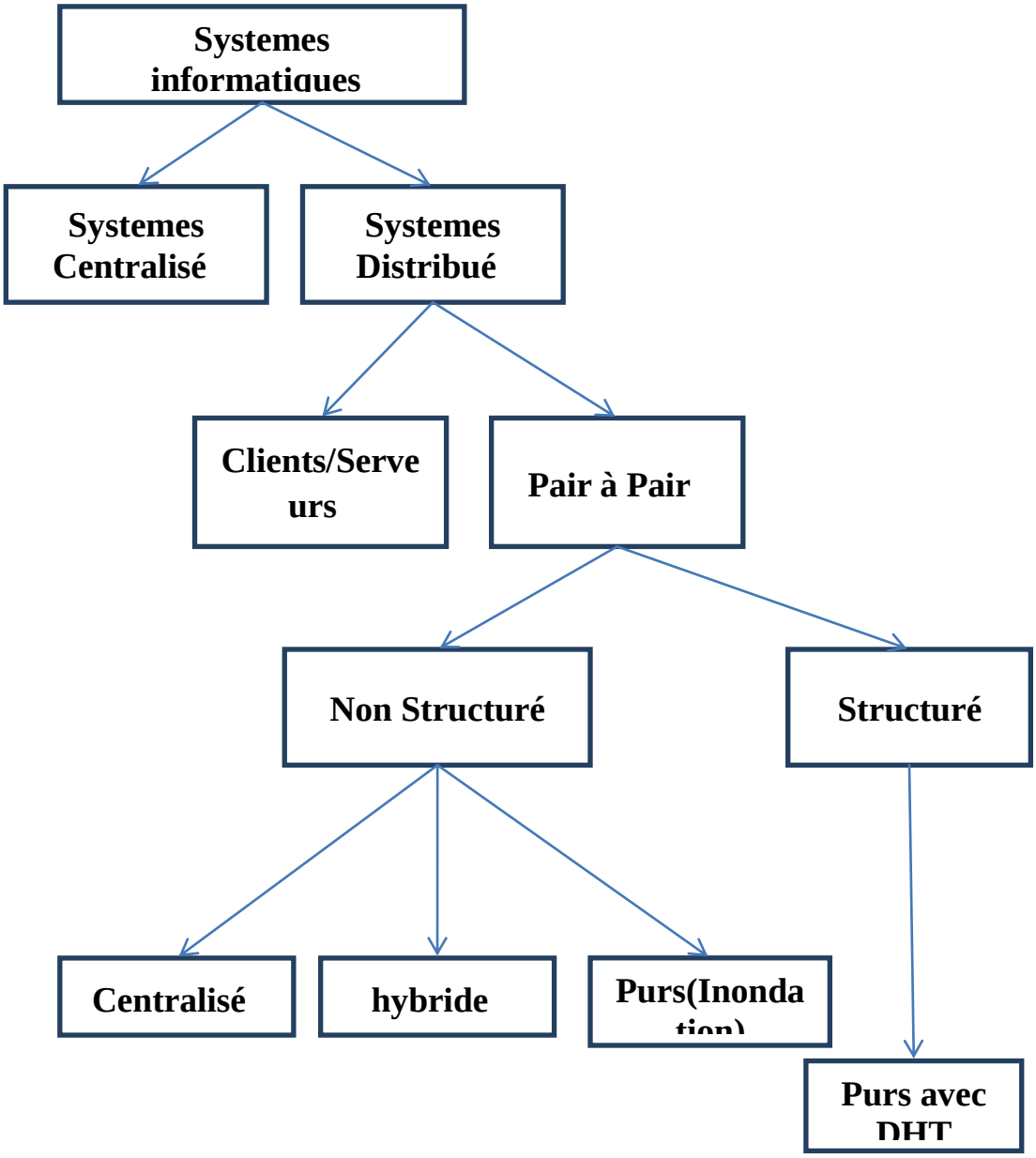


Figure 2.1: Classification des Systèmes distribués

On peut distinguer le modèle pair à pair soit Structuré ou non Structuré.

Non Structuré :

Definition : Un réseau pair à pair non structuré est exclusivement basé sur les liens physiques qui relient ces nœuds ceux-ci reposent sur une construction aléatoire du graphe de connexion. Ces réseaux n'ont aucune structure logique, Un nœud joint le réseau par l'intermédiaire d'un autre nœud déjà connecté.

Les mécanismes sont simples et faciles à implémenter cependant les performances lors de la recherche d'un pair sur le réseau sont souvent limitées. Un nœud désirant localiser une ressource, demande à ses voisins s'ils connaissent cette ressource, à leurs tours, ses voisins demandent à leurs voisins s'ils ont des connaissances de cette ressourcer et ce, jusqu'à une profondeur fixée par le système. Le nœud possédant la ressourcer renvoie une réponse qui parcourt le chemin initial dans le sens inverse.

Le réseau Gnutella est l'un des premiers réseaux non-structurés qui a mis en évidence l'abondance de communication entre pairs.

2.4.1. P2P décentralisé « purs » :

Dans l'architecture p2p décentralisé, on dispense des services des serveurs en autres termes on n'a pas besoin des serveurs, tous les nœuds sont égaux et jouent le même rôle, lorsque un pair est supprimé du réseau, les services offerts ne seront pas affectés (voir figure 2.2, par contre, l'absence d'un serveur central ayant une vue globale sur la localisation des ressources hébergées par les pairs dans le réseau P2P pose le problème suivant: Comment un pair peut découvrir et accéder à une ressource dans ce contexte ? Pour cela deux solutions ont été proposées : la première basée sur la technique de l'inondation et la deuxième est basé sur les tables de hachages distribuées "DHT". Ces deux solutions sont connue sous les noms des réseaux P2P décentralisés non structurés et réseaux P2P décentralisés structurés.[11][12]

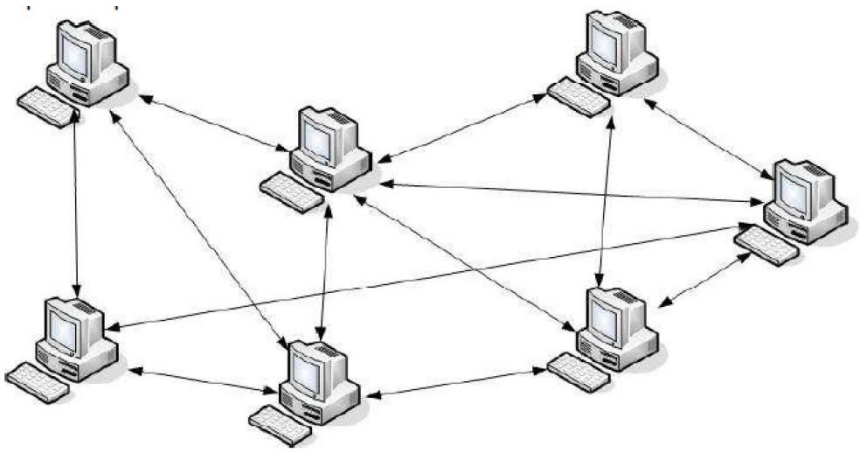


Figure 2.2: Architecture p2p décentralisé

2.4.1.1. Recherche par inondation:

Pour découvrir une ressource, une requête sera transmise d'un pair à un autre jusqu'à atteindre le client qui partage l'objet désiré, comme ici la technique est l'inondation des messages alors on peut vite tombé dans le cas des saturation des réseaux(bande passante) car on aura trop des messages échangé entre les pairs mais aussi les messages zombies et pour remédier à ces problèmes de l'inondation du réseau et messages zombie durant un temps trop long, le système associe à chaque requête un temporisateur TTL "Time To Live".

La valeur attribuée au TTL est généralement 7. Lorsqu'il arrive à zéro, la requête n'est plus renvoyée.

Oui le temporisateur TTL a pu régler le problème de l'inondation et de le message zombie mais par contre il a crée un inconvénient majeur de ce mécanisme, que est l'expiration du TTL car si le TTL arrive à zéro certes qu'on n'aura pas des problèmes des saturations mais on ne sera pas sûr de parcourir l'intégralité du réseau, ce qui peut aboutir à l'échec d'une recherche bien que l'objet désiré soit disponible sur le réseau P2P. Ce réseau passe mal à l'échelle et génère une surcharge du réseau par les trames de broadcast diffusées. Cette méthode est utilisée dans le protocole Gnutella.

2.4.1.2. Gnutella:

L'exemple de l'utilisation du réseau P2P décentralisé non structuré est gnutella, il est constitué d'un ensemble de pairs joignant le réseau d'après certaines règles (voir le tableau), la technique utilisé en gnutella est l'inondation alors nous avons quelques requêtes à prendre en compte exemple lors qu'un pair souhaite utilisé le réseau Gnutella. Premièrement un message Ping sera envoyé avec le but d'identifier les nœuds présents sur le réseau, on appelle ce message le message d'identification. ce message *Ping* sera envoyé à ses voisins, qui l'envoient à leur tour à leurs voisins et ainsi de suite, mais au bout d'un temps ce message ne sera plus renvoyer et sera stoppée alors dans ce cas c'est tout simplement à cause du TTL qui sera à 0. Alors l'autre requête à prendre en compte est le pong qui est une réponse a un Ping alors ici c'est très simple un client que reçoit un *Ping* sa réponse c'est automatique il répond avec le message *Pong* contenant l'adresse IP, le numéro de port, le nombre et la taille des fichiers partagés. Pour chercher une ressource dans le réseau, le client envoie une requête *Query* en spécifiant le nom de ressource et les critères de recherche, un servent qui reçoit ce message de type *Query* et s'il dispose de la ressource, renvoie une réponse *QueryHit* au voisin qui lui a retransmis la requête, spécifiant son adresse IP et son numéro de port TCP où l'objet peut être téléchargé. La réponse remonte de proche en proche jusqu'au client initiateur. Ce dernier télécharge ensuite en envoyant directement une requête de téléchargement au pair possédant le fichier. Un message *Push* est utilisé si les données sont derrière un firewall. [11][12]

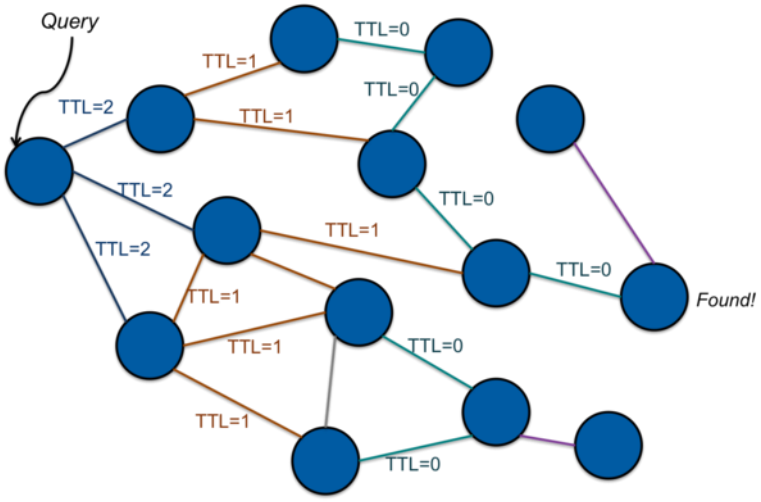


Figure 2.3: fonctionnement du Gnutella

Parmi les applications qui implémentent le protocole Gnutella, on trouve Limewire, BearShare, Gnucleos ou Phex, de plus les échanges peuvent être effectués quelque soit laplate forme(Windows, Linux/Unix, Macintosh, etc.) du client.

Type	**Description**	**Information**
Ping	Une requête à la recherche du pair	Vide
Pong	Réponse à un Ping	IP numéro du port nombre du fichier partagé
Query	Requête	IP numéro du port bande passante, nom du fichier
QueryHit	Une réponse à un query si on possède le ressource	IP, numéro du port, bande passante, taille de fichier
Push	Demande de téléchargement en cas de présence d'un firewall	IP, Index du fichier demandé numéro du port etc.

Tableau 2.1 : les requêtes gnutella

2.4.2.P2P centralisé :

Ici on a encore les serveurs avec une fonction un peu particulier et différent des serveurs du modèle client/serveur, ce modèle du p2p centralisé est la première génération P2P car le modèle p2p a fait son apparition par le concept de centralisation.il existe un serveur central, avec un objectif claire qui est d'avoir le rôle d'un annuaire qui va stocker que des informations concernant la description des ressources partagées (nom, taille,……) et l'emplacement d'où ils peuvent être pris, en d'autres terme les informations sur les utilisateurs qui les hébergent (nom utilisé, IP, nombre de fichiers partagés,……), comme illustré dans la figure ci-dessous.

Tout les nœuds souhaitant partager des ressource, alors il contacte le serveur central et lui déclare, celui-ci stocke son adresse IP ainsi un numéro de port donné par le nœud où il pourra être contacté pour un téléchargement. Quand un utilisateur a envie de faire une recherche d'un fichier, alors lui aussi prend contact avec le serveur central en lui envoyant une requête et celui-ci lui répond en transmettant la liste des nœuds possédant le fichier demandé leur IP leur numéro de port, alors l'utilisateur a le choix de choisir parmi les réponses indiquées par l'index central celle qui lui convient le mieux, il contacte directement le ou les postes choisis, Le contenu reste toujours du côté client, ne passant jamais par le serveur.

Le modèle centralisé est déterministe et il permet une recherche simple, il est facile à administrer et à contrôler, c'est le cas souvent où on a le serveur, il est peu coûteux, nécessite qu'un seul serveur central pour la découverte et une machine hôte pour l'accès à la ressource, cependant il présente plusieurs inconvénients, il n'est pas robuste car la surcharge ou la panne du serveur central rend tout le réseau indisponible, mais il passe très mal à l'échelle. L'exemple le plus connu reposant sur cette architecture est "Napster". [11][12]

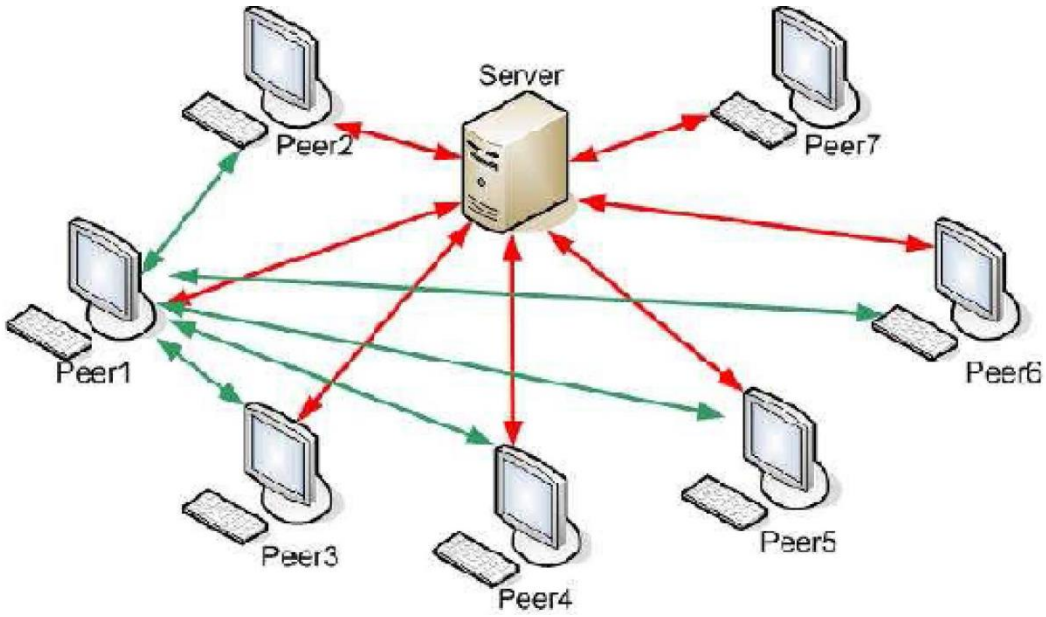


Figure 2.4: p2p centralisé

2.4.2.1. NAPSTER

Lors d'apparition du réseau P2P on a comme tout premier logiciel Napster, il est l'exemple de l'architecture P2P centralisé, alors sa façon de fonctionner c'est la façon standard de l'architecture P2P centralisé dont on a parlé précédemment ça veut dire que tous les membres du réseau doivent passer par le serveur qui est notre annuaire dans le but de l'informer des fichiers dont ils disposent mais aussi de le contacter pour l'obtention des coordonnées d'un élément lorsqu'on souhaite lancer une recherche des fichiers dans ce réseaux. Un Client désirant partager des fichiers doit exécuter sur son ordinateur le logiciel Napster. Etant connecté à Internet, le client établit une connexion TCP avec le serveur central Napster et lui déclare les fichiers partagés, le serveur central Napster qui est notre annuaire contient l'index avec toutes les adresses IP des clients participants, ainsi qu'une liste de ressources partagées, comme nous avons dit précédemment, le fichier ne transite pas par le serveur central. Le logiciel permet à l'utilisateur de se connecter au pair désiré directement, c'est cette communication directe entre les pairs qui différencie le modèle P2P centralisé du modèle client-serveur classique. [11][12]

- **Principe :**

a) Tout le monde doit avoir le logiciel Napster. On doit avoir la connexion internet lors de l'exécution du logiciel.
b) Après la stabilisation de la connexion entre le serveur et le client. Alors on déclare notre ressource, adresse IP, Port où on pourra être contacté.
c) Le serveur central maintient un répertoire des ordinateurs clients connectés et stocke les informations sur ces utilisateurs (notamment les fichiers en partage).
d) Le nœud voulant une ressource dans Napster doit contacter notre fameux serveur.
e) Le serveur renvoie à l'utilisateur une liste des réponses éventuelles : adresse IP, nom d'utilisateur, taille du fichier, l'utilisateur choisit le fichier qu'il veut et établit une connexion directe avec l'hôte du fichier, en lui envoyant sa propre adresse IP et le nom du fichier demandé.
f) Transfert du fichier entre les deux ordinateurs, puis connexion interrompue à la fin du transfert.

2.4.2.2. Bilan : avantages et limites

- **Avantages d'un système centralisé :**

– Présence d'un serveur central : facile à administrer, et donc facile à contrôler

– Evite les recherches coûteuses sur le réseau : pas de routage et planification de la gestion des utilisateurs

- **Les limites :**

– Pas d'anonymat partout car chaque pair est connu du serveur et des pairs sur lesquels il télécharge.

– Limites habituelles d'un serveur central : problème de disponibilité, de passage à l'échelle (saturation de la bande passante et du nombre de processus).

Cas de Napster : facile à fermer.

2.4.3.P2P hybrides :

Dans les hybrides on a un véritable mélange et une combinaison des caractéristiques entre les modèles Pair a Pair totalement décentralisé et les modèles Pair a Pair centralisé. Cette

solution a permis de surpasser quelques problèmes rencontrés dans le modèle centralisé, car son coté décentralisation assure l'extensibilité, la tolérance aux pannes et le passage à l'échelle.

Ici on a les Super Pairs ces derniers ont des rôle particulier, généralement ces sont des pairs qui ont une forte capacité de calcul et une large bande passante, alors ces super-pairs ont des groupes des autres pairs moins puissants où il sert comme le serveur local pour ce groupe de pairs, comme la montre la figure. De nombreuses applications sont construites selon ce modèle, par exemple : Kazaa, BitTorrent. [11][12]

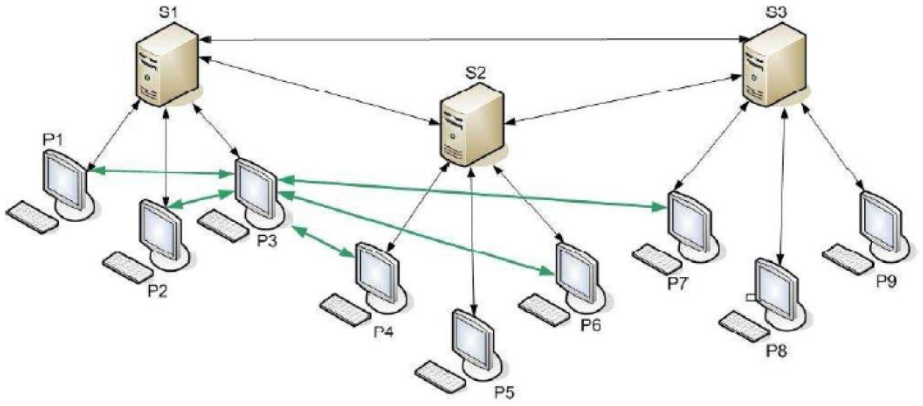


Figure 2.5: modèle p2p Hybride

Remarque:

Certaines classifications des architectures P2P considèrent les modèles hybrides et centralisés comme identiques : le modèle centralisé est un modèle hybride avec un seul super pair.

2.4.3.1. KaZaA:

Premièrement on doit classifier les pairs qui ont des capacités supérieurs aux autres pairs dans certains cas par exemple : une grande vitesse de connexion, une grande capacité de disque, et une grande capacité de traitement. Ceux-ci sont immédiatement désignés comme des super-pairs, ces derniers en plus de leur rôle spécifique à savoir l'hébergement de la liste des fichiers partagés par les clients peuvent aussi partager et télécharger des fichiers comme les autres pairs ordinaires. Tous les nœuds entre en contacte avec des super pairs

dont ils font partie pour pouvoir se connecter au réseau, et lui envoie la liste des fichiers qu'il désire partager afin qu'ils y soient indexés. Il lui envoie également des requêtes sur des fichiers qu'ils veulent obtenir. Une fois les contacts est établis entre les supers pairs et les pairs ordinaires alors les pairs obtiennent des adresses IP du pair qui dispose la donné recherche, le super pair qui fournie ce adresse IP peut soit la cherche dans sa propre indexation local(les liste des ressource hébergés) donc ici on est dans le cas centralisé ou bien, il passe des requêtes aux autres super-pairs donc la technique utilisée entre les super pair pour communiquer est l'inondation de messages, les données restent toujours distribuées sur les pairs et les échanges se font directement d'un pair à un autre via le protocole HTTP. [11][12]

2.5. Réseaux structuré

La deuxième grande classe est celle des réseaux P2P structurés, ils sont dits structurés, car au-dessus du réseau physique sous-jacent, les nœuds sont reliés par un réseau recouvrant construit sous certaines contraintes, répondant à plusieurs propriétés et connectant les peers selon une structure particulière donnée exemple en anneau : Chord ou cartésiennes : CAN.

On déploie une organisation de la topologie virtuelle dans l'architecture P2P structuré pour les méthodes de routage et localisation en vue de la faire correspondre à une topologie connue (anneau, can,...). Chaque topologie présente des méthodes de nommage, routage et localisation qui lui sont propre. Les principales études sont basées sur les tables de hachage distribuées (Distributed Hash Table). L'avantage principal de ces topologies est qu'elles garantissent de trouver la donnée une fois elle est présente dans le système.

2.5.1.Chord

Chord est un protocole de recherche distribué, qui repose sur une structure en anneau, Chord utilise les hachages pour assigner aux nœuds et aux données leur identifiants à m bit et les id sont compris entre 0 à 2^m-1 (m est la taille d'un identifiant). L'identifiant d'un pair

est un hachage réalisé à partir de son adresse IP et l'identifiant d'une ressource est le hachage de la donnée stockée.[13]

La figure suivante montre que les nœuds dans le protocole Chord sont ordonnés dans un cercle :

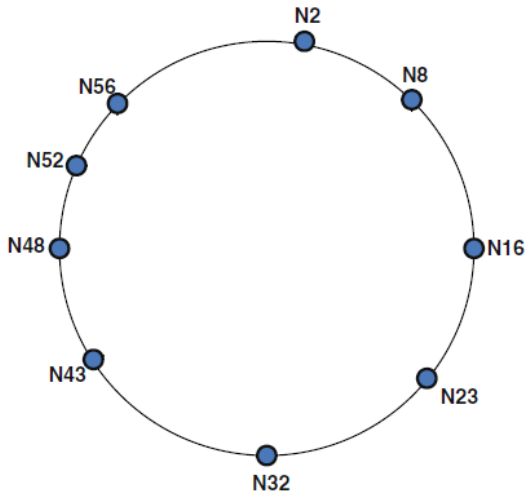


Figure 2.6: architecture Chord

Table de Routage

Ici on a deux parties dans les mécanismes pour les routages en Chord la premier tables de routage : les tables de routages (Finger Table) sont très importantes pour les mécanismes de recherche des ressources qui sont réparties sur les différents nœuds de l'anneau. Ces tables de routages ont m entré où m est la longueur de l'identifiants exemple étant donné que l'id dans nœud est 8, l'i^{eme}entré dans le table de routage pointe vers le nœud qui est le plus proche numériquement a $8+2^{i-1}$ dans la direction des aiguilles d'une montre. Exemple on suppose que 8 est l'id d'un nœud alors la 3^{eme} entrée dans le table pointe vers le nœud plus proche à $8+2^{3-1}=12$ dans ce cas 16, et la deuxième est la liste de successeur d'un nœud qui contient non seulement le successeur immédiat mais une liste des n-1 nœuds successeurs, alors ça permet de gérer la robustesse et l'arrivé et départ des nœuds comme présenté dans les figure suivantes.

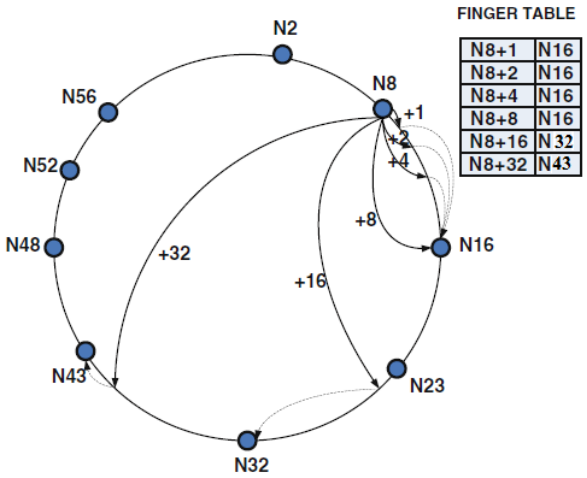


Figure 2.7: illustration table du routage

Arrivé et départ des nœuds

Pour un nouvel arrivant la première chose à faire est le boostraping qui est tout simplement une opération qui doit être faite pour tous les nouveaux arrivants. Le boostraping commence depuis l'arrivé du nœud jusqu'au fonctionnement normal de ce nœud : quelques étapes pour réalisé le boostraping

1) Le nœud arrivant doit utilise le hachage pour générer son id
2) Il doit contacter un nœud déjà existant dans les réseaux pour chercher son successeur.
3) Le nouveau nœud utilise le protocole de la stabilisation pour corriger les tables des routages ce protocole est utilisé périodiquement en arrière-plan ce protocole a deux fonction suivant :

   a) stabilise : que permet aux nœuds de savoir qu'il y a un nouvel arrivé ou un départ de nœud.

   b) Fixfingers : assure que les tables soit maintenue correctement
4) Construction de sa table de routage.

2.5.2.Pastry

Définition :

Pastry est un protocole qui réalise une table de hachage distribuée (DHT) similaire à Chord sur un réseau pair à pair organisé en un anneau virtuel de nœuds. Chaque nœud gère les informations dont les clés sont numériquement proches de son propre identifiant, sachant que les espaces d'identifiant de nœuds et de clés sont confondus dans ce protocole (dans le même espace).

Le protocole admet que des nœuds puissent rejoindre ou quitter l'anneau a tout moment.

Il doit néanmoins faire suivre toute requête pour une clé, provenant de n'importe quel Nœud, au nœud qui est responsable pour cette clé et qui répondra par l'information Recherchée, ce protocole réalise un system complètement décentralise, scalable, fiable.

Fonctionnement:

Le fonctionnement de Pastry(DHT) est presque identique à d'autres DHTs, Cela permet à Pastry de réaliser l'évolutivité et la tolérance aux pannes comme tous les autres réseaux, tout en réduisant le coût global de l'acheminement d'un paquet d'un nœud à un autre en évitant la nécessité d'inonder le réseau. Parce que la métrique de routage est fournie par un programme externe basée sur l'adresse IP du nœud cible, La métrique peut être facilement passée à un plus petit nombre de sauts.

L'espace des clés est pris pour être circulaire, comme l'espace des clés dans le protocole Chord et les identifiants des nœuds sont des entiers non signés 128 bits représentant position dans l'espace circulaire. L'ID des nœuds sont choisis aléatoirement et uniformément.

Les nœuds qui ont des IDs adjacents sont géographiquement dispersés.

Le réseau de recouvrement de routage est formé au dessus de la table de hachage par chaque pair en découvrant et en échangeant des informations d'état comprenant une liste de nœuds feuille, une liste de voisins, et une table de routage.

La liste de nœuds feuille comprend les L/2 pairs les plus proches à l'ID du nœud dans chaque direction autour du cercle.

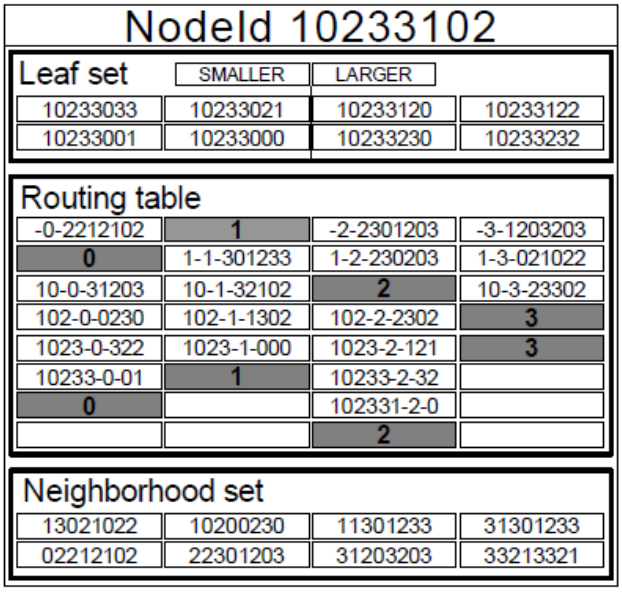
En plus des nœuds feuille, il y a aussi la liste de voisinage. Cela représente les M pairs les plus proches en termes de la métrique de routage. Bien qu'il ne soit pas utilisé directement dans l'algorithme de routage, la liste de voisinage est utilisée pour maintenir les directeurs de la localité dans la table de routage.

Enfin, il y a la table de routage elle-même. Elle contient une entrée pour chaque bloc d'adresses qui lui est attribué. Pour former les blocs d'adresses, la clé de 128 bits est divisée en chiffres avec chaque chiffre formé de b bits, ce qui donne un système de numérotation avec la base$\boldsymbol{2^b}$. Cela partitionne les adresses en niveaux distincts du point de vue du client, le niveau 0 représentant un préfixe commun à zéro chiffres entre deux adresses, niveau 1 un préfixe commun à un chiffre, et ainsi de suite. La table de routage contient l'adresse du pair le plus proche connu pour chaque chiffre possible à chaque niveau de l'adresse, sauf pour le chiffre qui appartient au pair lui-même à ce niveau particulier. Cela se traduit par le stockage de $\boldsymbol{2^b - 1}$ contacts par niveau, avec le nombre de niveaux à l'échelle de $\boldsymbol{(log 2N)/b}$. Les valeurs de $\boldsymbol{b \approx 4}$, $\boldsymbol{L \approx 2^b}$ et $\boldsymbol{M \approx 2^b}$ représentent des valeurs opérationnelles sur un réseau typique. [14]

- **Mécanisme de routage :**

Un paquet peut être routé à une adresse dans l'espace de clés s'il y a un pair ou pas avec cet ID de noeud. Le paquet est routé vers sa place sur l'anneau circulaire et le pair dont l'ID est plus proche de la destination souhaitée. Chaque fois qu'un poste reçoit un paquet ou veut envoyer un paquet, il examine d'abord son ensemble de feuilles et l'envoie directement au bon nœud s'il s'y trouve. Si cela échoue, le prochain pair consulte sa table de routage dans le but de trouver l'adresse d'un nœud qui partage un préfixe plus long avec l'adresse de destination que le pair lui-même. Si l'homologue n'a pas de contacts avec un préfixe plus long ou le contact est mort il reprendra un pair de sa liste de contact avec le même préfixe dont l'ID est numériquement plus proche de la destination et envoie le paquet à ce pair. Comme le nombre de chiffres corrects à l'adresse doit toujours augmenter ou rester le même, le protocole de routage fini toujours par converger.

Pour router les messages, Pastry utilise trois composants : une table de routage (liens vers des nœuds lointains), un ensemble de voisins (neighborhood set) (liens vers des nœuds proche géographiquement ou physiquement basés sur la latence du ping ou le nombre de sauts) et un ensemble de feuilles établies (leaf set) (liens vers des nœuds proches numériquement). Chaque donnée est indexée dans le nœud le plus proche numériquement. La figure suivante présente les tables du nœud 10233102 dans Pastry :



NodeId 10233102

Leaf set

SMALLER		LARGER	
10233033	10233021	10233120	10233122
10233001	10233000	10233230	10233232

Routing table

-0-2212102	1	-2-2301203	-3-1203203
0	1-1-301233	1-2-230203	1-3-021022
10-0-31203	10-1-32102	2	10-3-23302
102-0-0230	102-1-1302	102-2-2302	3
1023-0-322	1023-1-000	1023-2-121	3
10233-0-01	1	10233-2-32	
0		102331-2-0	
		2	

Neighborhood set

13021022	10200230	11301233	31301233
02212102	22301203	31203203	33213321

Figure 2.8: table du routage de pastry

Ajout d'un pair de nodeID X :

Remarque : on suppose qu'il connait un pair A proche de lui

- X demande le routage de la requête « join » de la clé X

  Remarque : routage jusqu'à pair Z, dont le nodeID est plus proche du numéro de X.
- En réponse les paires traverses envoient leur table d'état à X a partir desquelles il va construire la sienne (demandes possibles d'autres infos de X)

  1- comme A est supposé proche de X, l'ensemble de voisins Ma sera utilise pour initialise Mx

2- Z est lepaire dont le nodeIDestle plus proche numériquement, donc Lz utilise pour construire Lx.

3- Si requête X route ver B et B et X partagent même préfixe donc Rx utilise Rb

- X transmet une copie de son état a tous paire.

Départ/panne d'un pair de nodeID X :

Remarque : on suppose la panne d'un pair X si ses voisins (dans L) ne peuvent plus le joindre.

- Pour remplacer un pair dans L, son voisin contacte un pair avec le plus grand index du côté où le voisin est en panne, et demande son L
- Un pair se rend compte de la panne d'un pair via table de routage si ce dernier à besoin de le contacter.

  Le message peut être forwardé à un autre pair (pas de problème de routage) mais doit être remplacé pour préserver l'intégrité de la table de routage.

2.5.3.Le protocole CAN :

CAN ou Content Adressable Network, est l'un des premiers protocoles P2P basé sur les tables de hachage distribuées.

- **L'espace d'adressage :**

Comme tous les algorithmes basés sur les DHT, la localisation d'un nœud est calculée en utilisant une fonction de hachage. Après, l'espace total des coordonnées est divisée en « zones » où chaque nœud possède (est responsable) une zone.

CAN utilise un espace virtuel et d- dimensionnel de coordonnées cartésiennes pour traiter les nœuds et les données. Les données et les nœuds sont mappés aux points correspondants/coordonnées dans cet espace d-dimensionnel en utilisant une fonction de hachage uniforme. Donc, l'adresse d'un nœud est sa localisation dans le système de coordonnées. Un nœud responsable d'une zone est responsable des données qui se trouvent sur cette zone. [15]

- Mécanisme de routage :

Un nœud contient dans sa table de routage les coordonnées et les adresses IP de chacun de ses voisins. On classifie les nœuds comme voisins s'ils sont adjacents. On dit les deux zones sont adjacentes si les portées de leur coordonné se chevauchent quand on parle d'espace à d dimensions.

- La recherche des données :

La recherche est entamé avec la fonction de hachage de la donné recherché pour faire une correspondance à des coordonnés dans l'espace à d-dimensions, un message sera formé possédant les coordonnées de la destination et il sera transmit d'une façon gourmande vers la zone de destination. Le nœud transmet toujours le message vers son voisin qui est le plus proche de la destination. Voici une illustration de ce mécanisme.

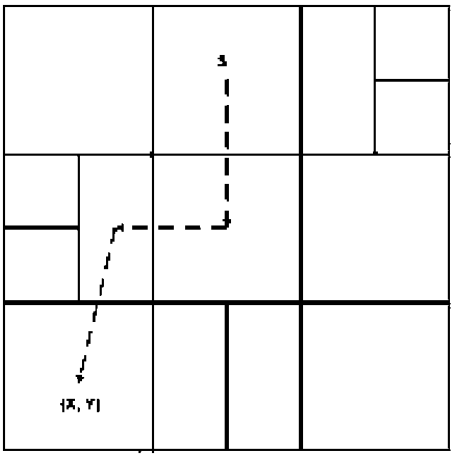


Figure 2.9: illustration du modèle CAN

- Arrivée et départ des nœuds :

Lors de l'arrivé d'un nœud dans CAN premièrement ce nœud doit connaitre déjà un pair qui existe dans les réseaux CAN après il doit générer ses propres coordonné P avec la fonction de hachage ensuite il envoie un message joint avec coordonné P alors le premier

nœud qu'il contacte va envoyer ce message a un nœud qui possède la zone de la coordonné P ensuite ce nœud va diviser sa zone en deux : alors le nouveau prend une partie et l'ancien reste avec une partie et sa table de routage sera construite a partir de l'ancien nœud qui possédait la zone ou il est affecté maintenant, et ensuite l'ancien et le nouveau nœud envoient des messages à leur voisin pour les informés et ce message sera aussi mise a jour pour leur voisins pour qu'ils puissent maintenir leur tables de routages correctement.

Le départ peut être normal ou bien d'une façon défaillant.

Un nœud qui décide de quitter le réseau CAN doit informer ses voisins de son intention alors on appelle ça un départ normal. Une fois qu'il quitte le réseau l'un des voisins prend la zone laissé par l'ancien nœud (de préférence le voisin qui a la plus petite zone).

Dans le cas contraire où un nœud est défaillant et quitte le réseau sans préavis, alors un message périodique de mise à jour découvrira ce départ ou cette panne. Alors le nœud qui a découvert la panne lance le mécanisme de prise de contrôle avec un compte à rebours qui sera démarré. Une fois le délai expiré, le message TAKEOVER sera envoyé à tous les voisins de nœud défaillants, ce message contient l'information sur la zone du nœud qui a découvert la panne. Le nœud qui reçoit ce message annule son propre compte à rebours au cas où sa zone est supérieure à la zone contenu dans le message TAKEOVER, ce message a pour but de donner la zone orpheline au voisin qui possède la zone la plus petite pour l'équilibre.

Dans CAN la table de routage est maintenue correcte grâce aux messages des mis-à-jour périodique envoyer entres le voisins.

2.6. Introduction à JXTA

Un des cadres bien connus pour le développement d'applications P2P est JXTA(Juxtapose) qui est un protocole P2P open source développé par Sun Microsystems en 2001. Dans ce protocole, tous les appareils qui sont connectés au réseau peuvent échanger des messages au format XML et de travailler ensemble, indépendamment de la topologie réseau sous-jacent. JXTA est basé sur le format XML et il peut être utilisé dans tous les langages actuels tels que: Java, C # et C ++. Le but du protocole JXTA est que les pairs peuvent

partager les ressources sans se soucier de la topologie du réseau. Par exemple, les pairs peuvent partager les ressources qui sont derrière un NAT ou pare-feu même en tant que pairs directs à Internet.

2.6.1. Origine de JXTA?

JXTA est né chez SUN Microsystem. JXTA est un projet Open Sourceet a été annoncé par Bill Joy 15 fevrier 2001.

Le site de JXTA a ouvert le 25 Avril 2001 hébergé par collabNet, il héberge une centaine de projets actuellement. Et il y a 17,445 membres Influents :

UNIX JAVA(1995) XML (1998) Systèmes P2P (1999) JXTA (2001)

JXTA est un projet de Sun R&D mené par Bill Joy et Mike Clary, l'objectif : trouver un nouveau type d'informatique distribuée. [16]

2.6.2. Les objectifs de JXTA ?

1- Interopérabilité (entre application, entre les différents systèmes P2P)

2- Indépendance des applications, langages, OS, réseaux.

3-Sécurité à différents niveaux, prise en compte dés le noyau jxta (JXTA permet de réaliser des applications pour les entreprises soucieuses de la sécurité) [16][17]

2.6.3. Les protocoles JXTA ?

Il y a seulement six protocoles dans le système actuel JXTA, chaque protocole est conçu pour le traitement des services JXTA.[16][17]

Ces six protocoles qui créent la couche du noyau JXTA sont énumérés ci-dessous:

Endpoint Routing Protocol (ERP) :

RendezVous Protocol (RVP) :

Peer Resolver Protocol (PRP) :

Peer Discovery Protocol (PDP) :

Peer Information Protocol (PIP) :

Pipe Binding Protocol (PBP) :

Chaque protocole est responsable de la gestion de services spécifiques dans la plate-forme JXTA et la collection de ces protocoles permettra de créer la couche de base de la plate-forme JXTA.

Peer Resolver Protocol (PRP) :

Ce protocole est utilisé pour envoyer une requête à un ou plusieurs pairs et recevoir la réponse à la requête. Les requêtes peuvent être envoyées à tous les pairs ou à un pair spécifique dans le groupe.

Peer Discovery Protocol (PDP) :

Dans ce protocole, les pairs peuvent annoncer leurs ressources ainsi que découvrir les ressources des autres pairs, par exemple: Peer groupe publicité, les services et les tuyaux. Au sein de ce protocole tous les pairs publient leurs ressources à travers le réseau JXTA. Les publicités ici sont représentées en tant que document XML et elles ont des métadonnées structurées qui décrivent les ressources des pairs.

Peer Information Protocol (PIP) :

Ce protocole permet d'obtenir les informations d'état des pairs tels que: la disponibilité, la charge de trafic, capacités et autres informations.

Pipe Binding Protocol (PBP) :

Grâce à ce protocole, les pairs peuvent établir une communication ou un tuyau avec un ou plusieurs pairs.

EndpointRouting Protocol (ERP) :

Le protocole Endpoint est responsable de la découverte de la route des pairs, donc les pairs peuvent découvrir une route ou des séquences de sauts qui seront utilisés pour envoyer des messages entre pairs. Par exemple, si le pair "Monsieur Benmamar" veut envoyer un message à "Monsieur Belhocine" et il n'y a pas de route directe entre "Monsieur

Benmamar" et «Monsieur Belhocine», alors le pair "Monsieur Benmamar" a besoin de trouver un(des) pair(s) intermédiaire(s) qui peut(peuvent) rediriger le message à "Monsieur Belhocine ". Endpointprotocol est également connu sous le nom de EndpointRouting Protocol (ERP) et il est utilisé pour détecter les informations d'itinéraire entre pairs. Par conséquent, s'il y a une modification de la topologie du réseau ou s'il devient indisponible, ERP peut être utilisé pour déterminer un itinéraire alternatif.

RendezVous Protocol (RVP) :

Protocole Rendezvous est utilisé pour propager des services par l'intermédiaire du groupe de pairs entier. Dans le groupe de pairs, les pairs peuvent être pair ou pairs Rendezvous qui écoute des pairs rendez-vous. Ce protocole permet à des pairs d'envoyer des messages à tous les cas d'écoute de services. Rendezvous pairs peut être utilisé par Peer Resolver et des pipes de liaison de protocoles afin de propager des messages à travers le groupe de pairs.

Architecture JXTA [17]

La plateforme JXTA est constituée de trois couches, comme représenté sur la figure

- Core Layer
- Service Layer
- Application Layer

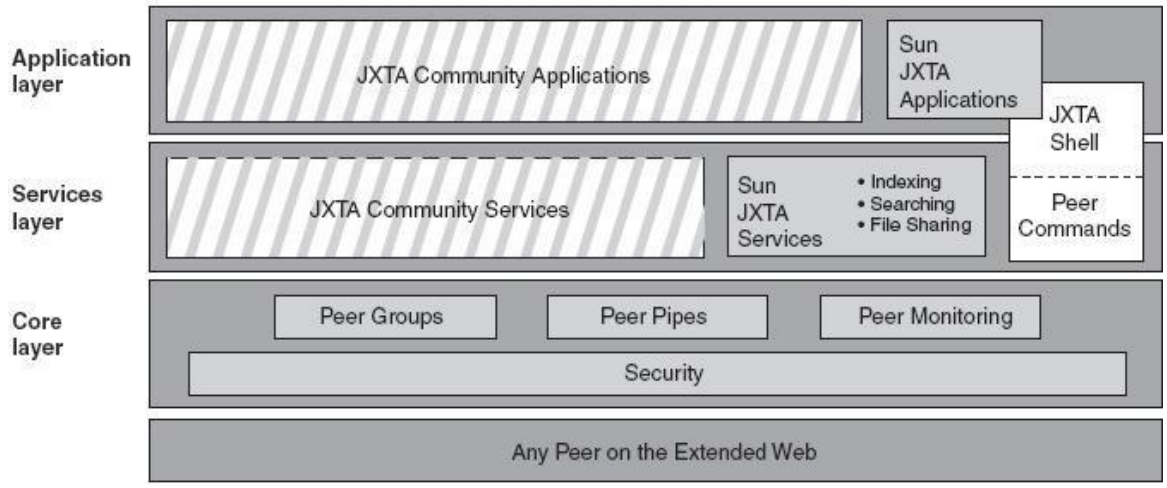


Figure 2.10: Architecture JXTA

Core Layer

La Core Layer (couche de base) est faite de tous les facteurs essentiels et communs des réseaux P2P, ces facteurs sont: Peers, Peer Discovery, la communication par les pairs et primitives de sécurité associés. Cette couche est partagée entre tous les dispositifs Peer-to Peer donc pour permettre l'interopérabilité.

Service Layer

Service Layer (couche des services) Cette couche est au dessus de la couche de base et est une couche commune pour les réseaux P2P. Exemples de services réseau inclut: Partage de fichiers, stockage.

Application Layer

Ici on a une liaison forte avec service layer, elle est utilisée quelque fois pour créer les autre services.

2.7. Conclusion :

Dans ce chapitre nous avons parlé des différents architectures des réseaux p2p, nous avons vu trois types de réseaux non structurés dont le centralisé, l'hybride et le décentralisé où

nous avons parlé de leur qualité et défauts, de même nous avons mentionné trois types de réseaux structurés et l'amélioration qu'ils apportent par rapport aux réseaux non structurés tout en énumérant quelque inconvénient de ces derniers.

Enfin nous avons fait une petite introduction sur JXTA et les ensembles de ses protocoles.

3.Chapitre 3: Présentation de l'application

3.1. Introduction

Apres avoir parlé de jxta sur lequel est basé ce projet, dans ce chapitre je vais parler de l'objectifs de cette application les caractéristiques, comment cette application est réalisée. Je vais décrire quelques classes utilisées lors de sa création mais aussi je parlerai du fonctionnement du programme et on aura quelques images de l'interface de l'application.

3.2. Objectifs

Ce projet comprend le développement d'une application de partage de fichiers peer-to-peer basé sur la plateforme JXTA. Il vise à donner aux utilisateurs la capacité de partager leurs fichiers sur leurs propres réseaux P2P, il vise à aider les amis à partager leur fichier en toute sécurité.

3.3. Caractéristiques de l'application

1. Le téléchargement est réalisé pour tous les membres dans le groupe, chaque pair peut partager et téléchargé les fichiers.
2. Lors de la recherche d'un fichier on doit écrire le mot clé exact.
3. La recherche est déterministe.
4. Utilisation du Content Management Service(CMS) pour gérer les partages et les téléchargements.
5. Un pair peut télécharger un fichier en même temps qu'on télécharge les fichiers qu'il partage.

3.4. Réalisation de l'application

Pour la réalisation de cette application j'ai décidé de choisir Java comme langage de programmation. Pourquoi Java ? L'une des grandes forces de ce langage est son excellente portabilité : une fois le programme créé, il fonctionnera automatiquement sur tous les autres systèmes d'exploitation qui ont Java Virtual Machine (JVM) installée. Exemple : Windows, Linux, Mac. C'est le langage où j'ai beaucoup de connaissance et de maitrise par rapport aux autres langages.

Donc j'ai utilisé Netbeans pour me faciliter ce travail en particulier pour faire le design de l'interface graphique car il est plus simple de créer une interface graphique avec Netbeans par rapport à Eclipse.

Comme cette application est basée sur JXTA le programme doit avoir la librairie JXTA qui contient des paquets « .jar » qui sera ajoutés lors de la création du nouveau projet avec Netbeans, c'est cette librairie qui nous aidera à créer l'application. Parmi les paquets les plus importants et l'un des plus indispensable il y a le CMS (Content Management Service) dont je vais parler après. J'ai utilisé aussi des icônes pour rendre l'interface graphique plus jolie et plus attrayant.

3.4.1.Content Management Service (CMS)

Le Content Management Service est une extension des bibliothèques JXTA et il est utilisé pour fournir des fonctionnalités de base à des fins de partage de fichiers. Par exemple, cette bibliothèque contient des composants pour: le partage de contenu entre les pairs dans le groupe, réponse à des requêtes de contenu provenant d'autres pairs, le téléchargement de contenu d'autres pairs, des informations sur la progression du téléchargement et de nombreux autres composants qui dépassent les notions de ce mémoire. Ensuite je parlerai des composants qui ont été utilisés pour créer cette application. Je présenterai aussi quelques classes, chacune avec le code utilisé pour sa création et leurs explications respectives.

3.4.2.Entités JXTA

Il existe plusieurs entités qui sont communes dans les applications JXTA et qui permettent de créer des applications P2P.

Les entités comprennent [17]:

- Peer
- Identifiers
- Peer Group
- Advertisements
- Messages

- Pipes

Peer

L'élément le plus commun dans tous les systèmes P2P sont les pairs. Le Peer est un composant qui fonctionne sur l'ordinateur avec la capacité de communiquer avec d'autres pairs.

Pour créer un système P2P, il est fondamental que les pairs aient la capacité de communiquer les uns avec les autres. Les éléments essentiels dans le réseau JXTA sont listés ci-dessous: Peer identifier (ID), Peer member-ship, Peer transport – la communication avec d'autres pairs dans le groupe.

Identifier

Chaque ressource dans les réseaux P2P doit être référencé en quelque sorte, Dans le réseau JXTA, les entités peuvent être référencés par un ID unique; JXTA utilise 128-bit Universal Unique Identifier (UUID) pour identifier et référencer les ressource dans le réseau P2P.

Ces UUID sont générés par la plate-forme JXTA et sont garantis sans chevauchement en ajoutant les détails du réseau et le nom pour eux. Un exemple simple du groupe UUID est: "JXTA: uuid-4E0742B0E54F4D0ABAC6809BB82A311E02". Ces UUID peuvent être utilisés pour identifier les différentes entités du réseau P2P tel que: Peer, Module Publicité, Services, groupe de pairs ... etc.

Peer Groups

L'entité Peer Groups dans le réseau JXTA est composée des pairs qui coopèrent entre eux pour partager leurs ressources et services les uns avec les autres. Par défaut, tous les pairs sont membres de NetPeerGroup, qui est le groupe par défaut dans JXTA. Lorsque les pairs utilisent des protocoles JXTA, ils rejoignent d'abord le NetPeerGroup qui est un groupe racine dans JXTA et alors ils peuvent rechercher un groupe spécifié et rejoindre ou créer un nouveau groupe.

Advertisements

Les advertisements sont des ensembles de messages XML et métadonnées qui sont utilisés par des pairs pour annoncer leurs services à d'autres pairs au sein même du groupe. Quand un pair publie son advertisement sur le réseau P2P, d'autres pairs sont informés de ses ressources disponibles et des services qui sont destinés à être partagés sur le réseau.

Par exemple: advertisements de fichiers et pipe advertisements. Par conséquent, lorsque les pairs découvrent ces annonces, ils sauront quelle est la source du fournisseur et pourront établir la communication avec le propriétaire de l'annonce afin d'utiliser ses services ainsi que ses ressources.

Messages

Les messages sont des objets XML qui peuvent être échangés entre pairs.

Pipes

Dans le réseau JXTA, les tuyaux sont utilisés pour envoyer un message entre pairs.

Maintenant je vais parler de quelques classes que j'ai utilisées pour réaliser cette application du partage de fichiers. Parmi ces classes on a :

EcouteDesPaires.java,

PartageDuFichier.java,

RechercheDuFichier.java,

TelechargementDuFichier.java,

La classe EcouteDesPaires.java

Le but de cette classe est de lister les noms des pairs disponibles dans le Group. Pour ce faire, cette classe obtient d'abord des services de découverte de groupe actuel, puis elle va créer un objet à partir de l'interface dans la bibliothèque JXTA appelé "Discovery Event Handler". Cet objet est responsable de la mise à jour de la liste des pairs dès que l'annonce

d'un pair est reçue par des pairs dans le Group. Le code suivant présent la façon dont cela est fait.

```
// Obtenir les services de découverte du Group

myDiscoveryService = SaEeDGroup.getDiscoveryService();

// Création d'un objet écouteur de la bibliothèque JXTA

public void discoveryEvent(DiscoveryEvent event)

{DiscoveryResponseMsg res = event.getResponse();

PeerAdvertisementpeerAdv = res.getPeerAdvertisement();

 if(peerAdv != null){ name = peerAdv.getName();}

Enumeration en = res.getAdvertisements();

        while(en.hasMoreElements()){

myAdv = (PeerAdvertisement) en.nextElement();

peerList.addElement(myAdv.getName());}

// Mise à jour de la liste noms des pairs

updatePeerList(peerList);}
```

La classe PartageDuFichier.java

Cette classe va partager tous les contenus de chaque Peer avec les autres pairs du Group. Pour ce faire, la CMS a été utilisé. CMS est une bibliothèque d'extension JXTA qui est utilisée pour partager des fichiers. Cette classe va d'abord créer un objet de la bibliothèque CMS, alors cet objet CMS doit être initialisée ainsi, il peut lier ses services au groupe courant. Ensuite, cette classe va commencer à chercher les fichiers dans le répertoire partagé - une fois que les fichiers sont collectés dans le répertoire de partage, l'objet CMS crée un advertisement unique pour chaque fichier et les rendre disponibles pour d'autres pairs. De plus, si l'un des pairs dans le groupe actuel demande à télécharger le fichier partagé, alors CMS va gérer cette demande et envoyer le fichier. Il y a aussi une autre classe utilisée ici appelée "CheckSumCalc"- Responsable de générer la checkSum pour les fichiers partagés avec d'autres pairs dans le groupe. Par conséquent, lorsque les autres pairs téléchargent des fichiers, ils régénèrent le checksum du fichier en utilisant le même algorithme (CRC-32 algorithme utilisé pour générer 32 bits en valeur hexadécimale) pour vérifier que le téléchargement a réussi et que le fichier n'a pas été corrompu. Le code suivant montre l'utilisation du CMS afin de partager des fichiers:

```
//Creation d'objets CMS

cms = new CMS();

//Initialisation d'objets CMS

cms.init(Group,null,null);

// utilisation du Content Manager dans l'objet CMS pour le
partage de fichiers

contentManager = cms.getContentManager();

// Partage de fichiers dans le réseau actuel JXTA

if(list[i].isFile())
```

```
{// Partage de fichiers et de contrôle dans le réseau

contentManager.share(list[i],checkSum.getFileSum(list[i]));

}
```

La classe RechercheDuFichier.java

Le but de cette classe est de rechercher des contenus partagés par d'autres pairs dans le groupe courant JXTA. Une fois que cette classe est exécutée par l'utilisateur, le programme commence à chercher un nom de fichier spécifique jusqu'à ce que l'utilisateur interrompe la progression de la recherche. L'utilisateur peut télécharger des fichiers partagés avec les connaissances de base obtenu à partir de la publicité des fichiers dans Group. Le fichier contient l'advertisement des informations telles que: Peer propriétaire, id de fichier, taille du fichier et l'adresse JXTA du fichier. En acquérant ces informations, les utilisateurs peuvent télécharger le fichier avec d'autres classes - expliqué plus loin dans ce chapitre. Le code suivant montre une petite partie des codes qui ont été utilisés pour créer cette classe:

```
//cette classe utilise CachedListContentRequest comme super-
class

class ListRequestorextendsCachedListContentRequest

{

//initialization de la super class avec PeerGroup and nom de
fichier

//que nous cherchons.

public ListRequestor(PeerGrouppg , String SubStr){
```

```
super(PeerGroupGrouppg,SubStr);

}

//notifier l'application si les annonces de fichiers trouvés

//danspeergroup

public voidnotifyMoreResults()

{

log.append("[+]recherche des plus des contenues.\n");

for(inti=0; i<searchResult.length;i++){

log.append("[*]touvé: " + searchResult[i].getName()+"\n" +

            "Size:  "  +  searchResult[i].getLength()  +  "
Bytes\n");

}

 }

}
```

TelechargementDuFichier.java

Cette classe se chargera de répondre à la demande de téléchargement de fichier. Cette classe est une sous-classe de "GetContentRequest" dans la bibliothèque CMS. Ainsi, lorsque cette classe est exécutée par l'utilisateur, il va essayer de télécharger le fichier spécifié avec les informations de ce fichier dans le groupe actuel. Une fois le téléchargement du fichier terminé, le checksum sera régénéré pour le fichier téléchargé et comparé au checksum du fichier d'origine qui a été obtenu à partir de l'advertisements du

fichier pour nous assurer que le téléchargement a réussi et le fichier n'a pas été corrompu. Le code suivant réalise la fonction décrite.

```
//cette class estunesubclass du GetContentRequest

class GetRemoteFile extends GetContentRequest

{

public GetRemoteFile(PeerGroup group, ContentAdvertisementcontentAdv,
Filedestination)

{

this.progressBar = progress;}

// informer l'utilisateur a propos de la progression actuelle du téléchargement

public voidnotifyUpdate(intpercentage)

{progressBar.setValue(percentage); }

//informer l'utilisateur quand le telechargement est fini

public voidnotifyDone()

{

log.append("[+]le telechargement est fini avec succes.\n");}
```

3.5. Fonctionnement

Le fonctionnement est simple, une fois exécutée cette application présente les termes et conditions qu'on doit accepter. Ensuite il faut configurer le login dans la jxta configurateur où on peut choisir le pseudo-nom qui nous sera attribué dans le groupe, après on sélectionne le dossier qu'on désire partager et on publie leur contenue dans jxta, l'utilisateur a le choix de chercher toutes les sortes de fichiers qu'il souhaite et les télécharger si les fichiers existe.

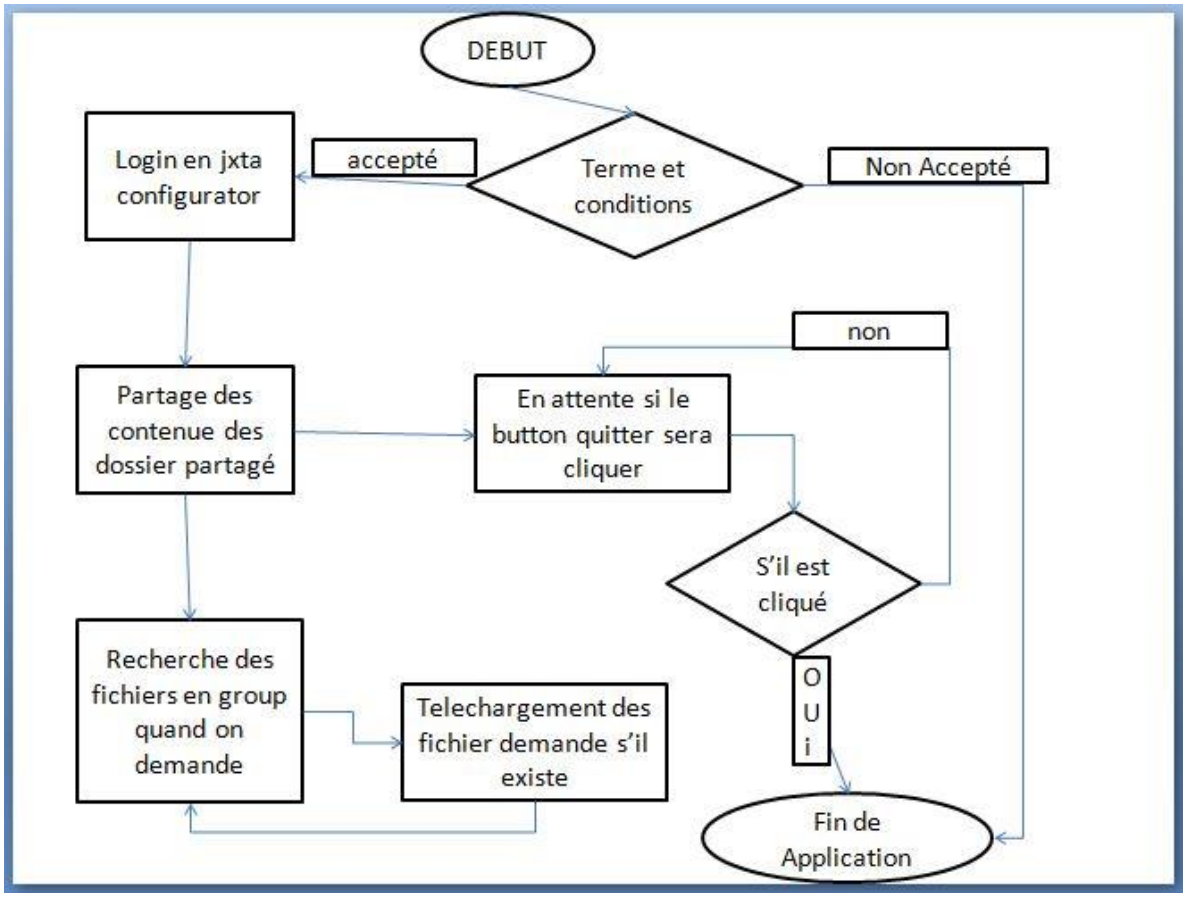


Figure 3.1: fonctionnement du jeu

3.6. Interface graphique

L'interface graphique d'une application est essentiellement faite pour faciliter son utilisation.

Le résultat final de ce projet a une fenêtre principale qui encapsule les fonctionnalités de tous les composants utilisés dans la présente application. Il est aussi le parent d'autres composants utilisés. La figure montre le châssis principal qui a été utilisé pour créer cette application. La partie suivante décrit la fonctionnalité de chaque élément dans l'unité centrale.

Dans notre application la première fenêtre que nous aurons est la fenêtre “a propos de moi” dans cette fenêtre nous aurons une petite description de l’application et aussi les termes et conditions, où on invitera les utilisateurs à respecter les conditions fixé pour utiliser cette application. Comme illustré dans la figure.

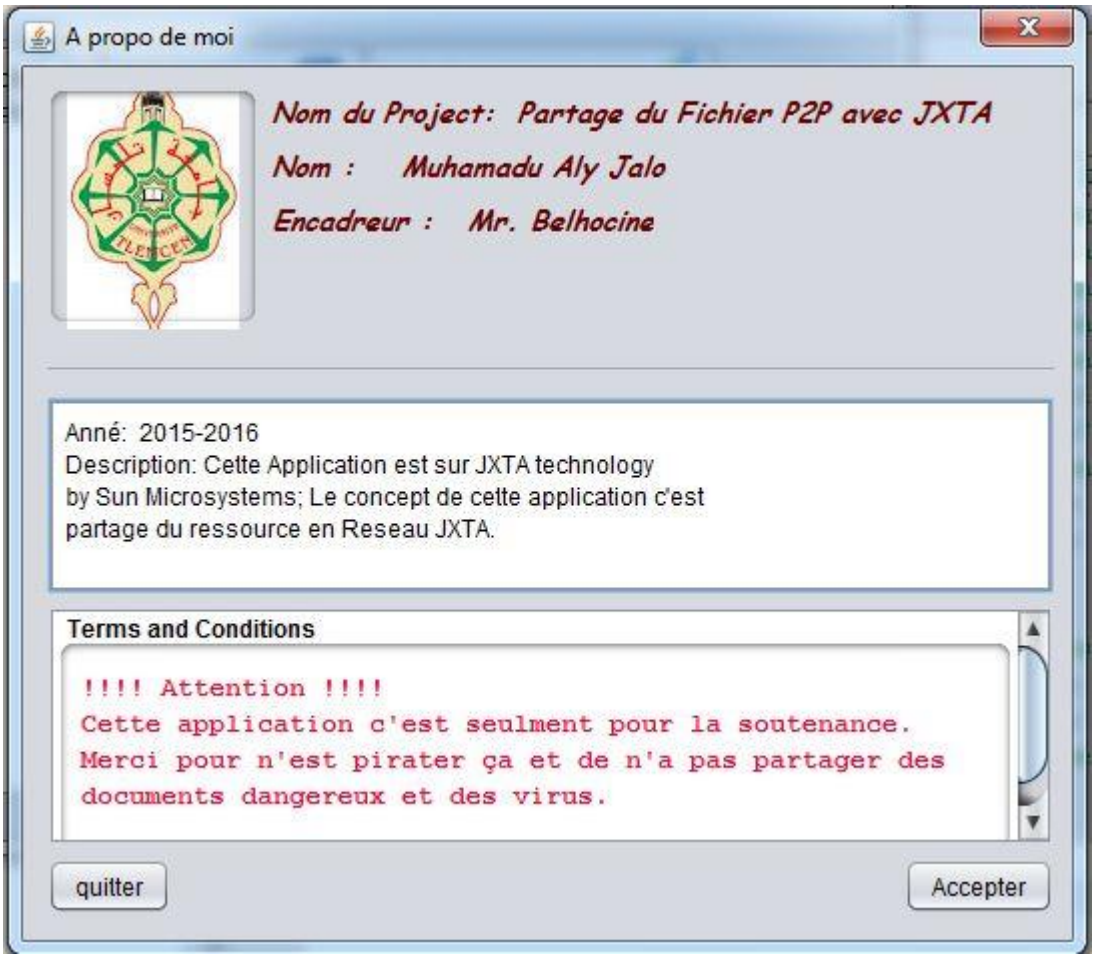


Figure 3.2: interface des termes et conditions

Une fois les termes et conditions accepter on a une fenêtre jxta configurateur qui nous apparaitra où nous pouvons nommer notre pc dans le réseau et aussi configurer notre adresse IP et le port où il sera disponible.

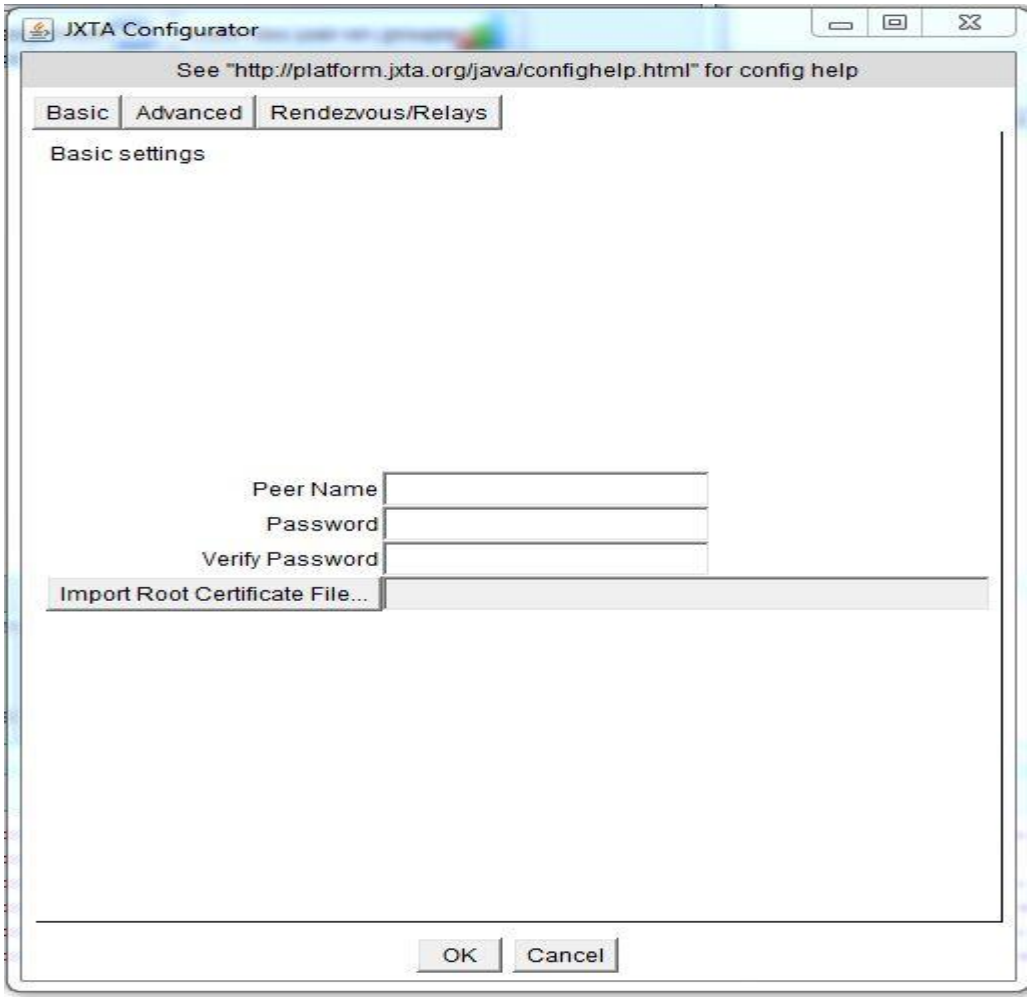


Figure 3.3: interface jxtaconfigurator pour le nom et mots de passe

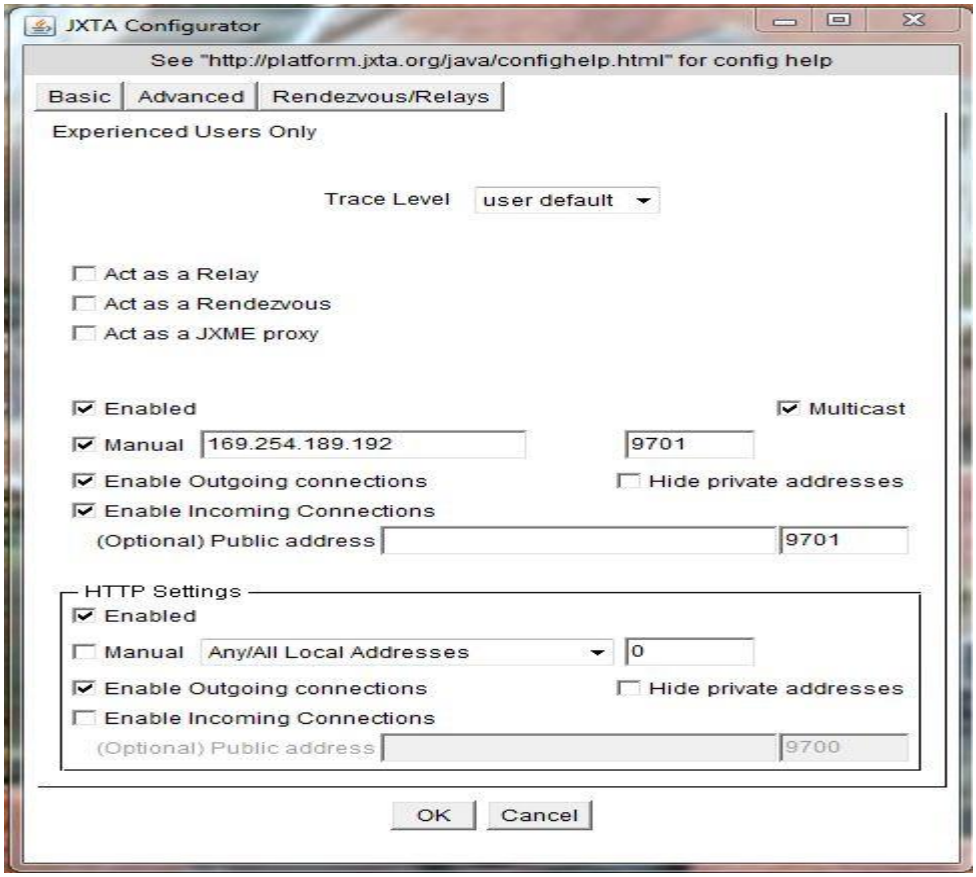


Figure 3.4: interface jxta configurator pour port et adresse IP

Une fois qu'on a configuré les adresses IP et le nom, il nous apparaitra une fenêtre qui nous invite à choisir le dossier qu'on veut partager.

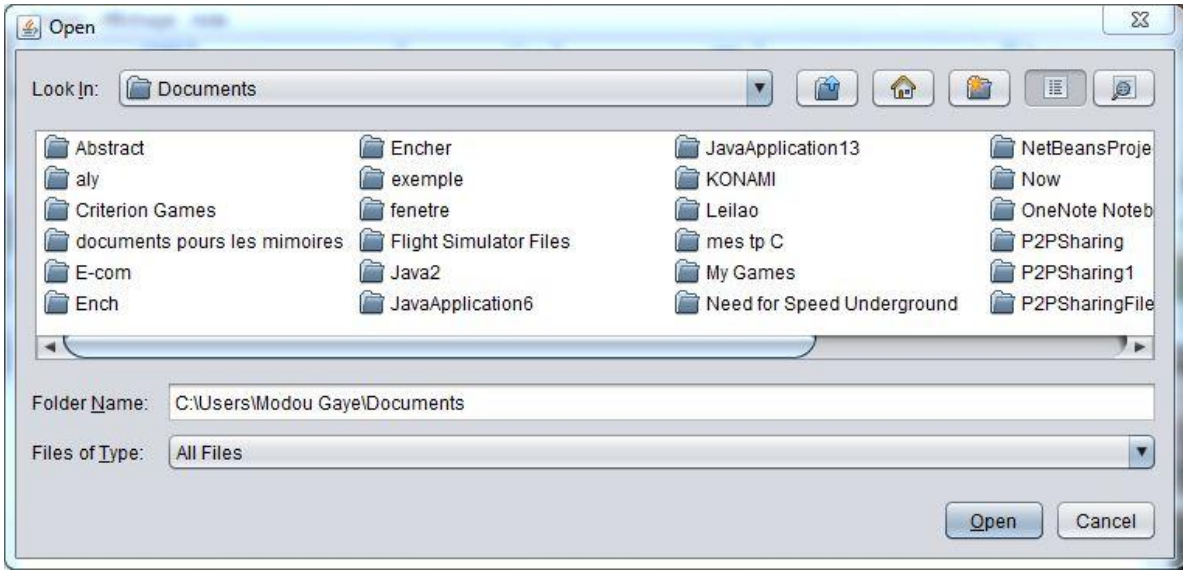


Figure 3.5: interface pour la sélection du dossier qu'on désire partager

Ce projet contient un menu qui est composé de trois éléments ou chacun représente une fonctionnalité différente:

1) Fichier

2) Affichage

3) Aide

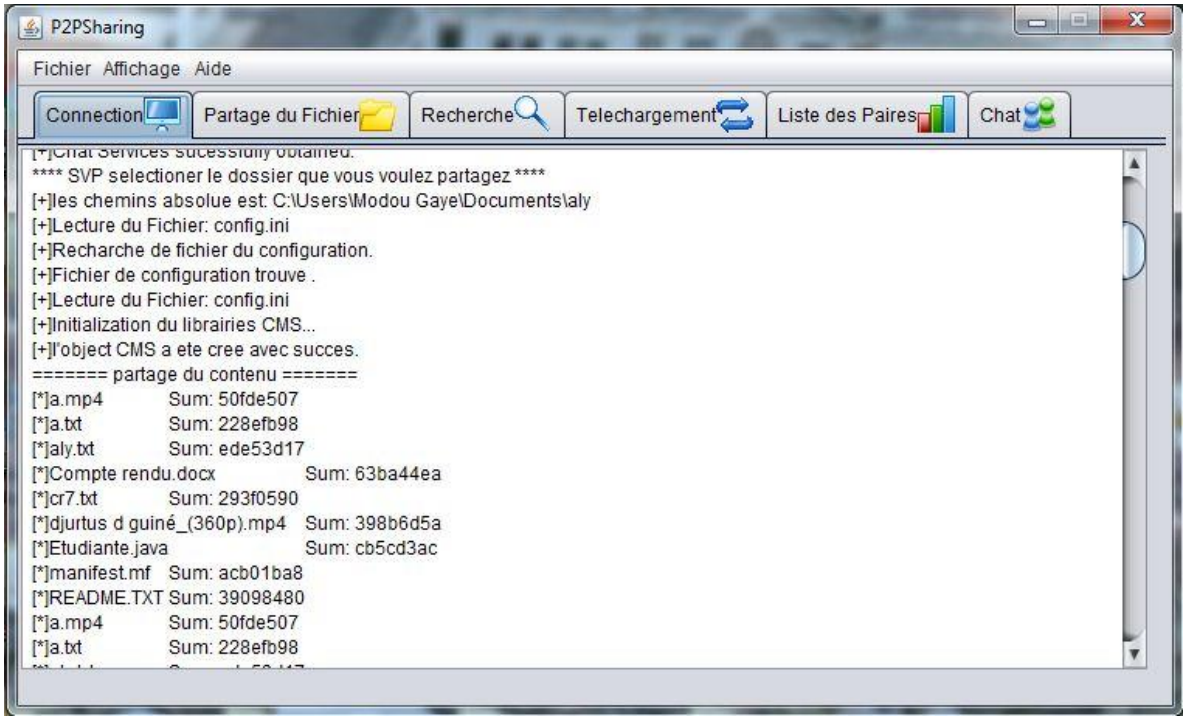


Figure 3.6: interface principale pour les informations du fonctionnement

Fichier

Avec l'option "Fichier", les utilisateurs peuvent se déconnecter du groupe de partage et se reconnecter au groupe de partage. En outre, l'utilisateur peut facilement sortir du programme principal. Une vue de cemenu est présenté dans la figure

Figure 3.7: le menu fichier

Affichage

Elément Affichage est conçu de telle sorte que l'utilisateur peut avoir différentes langue selon son besoin et sa compréhension, les langues tels que: l'Arabe, le Français et l'Anglais sont disponible comme illustré dans la figure suivante.

Figure 3.8: menu affichage

Arabe

Figure 3.9: L'option langue arabe

Remarque : la langue Arabe ne sera pas utilisée pour faire la recherche des fichiers, elle est utilisée simplement pour faciliter la compréhension de l'utilisateur.

Anglais

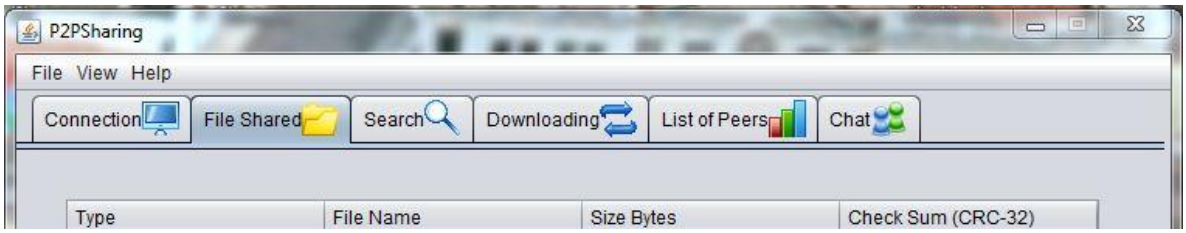


Figure 3.10: L'option langue anglaise

Partage du dossier

Ce panneau a été conçu de telle sorte que les utilisateurs peuvent visualiser leurs fichiers partagés et téléchargés. Ceci inclut également des informations sur les fichiers dans le dossier partagé tels que: Type de fichier, Nom de fichier, la taille du fichier en octets et fichiers checksums. Cet onglet est représenté sur la figure.

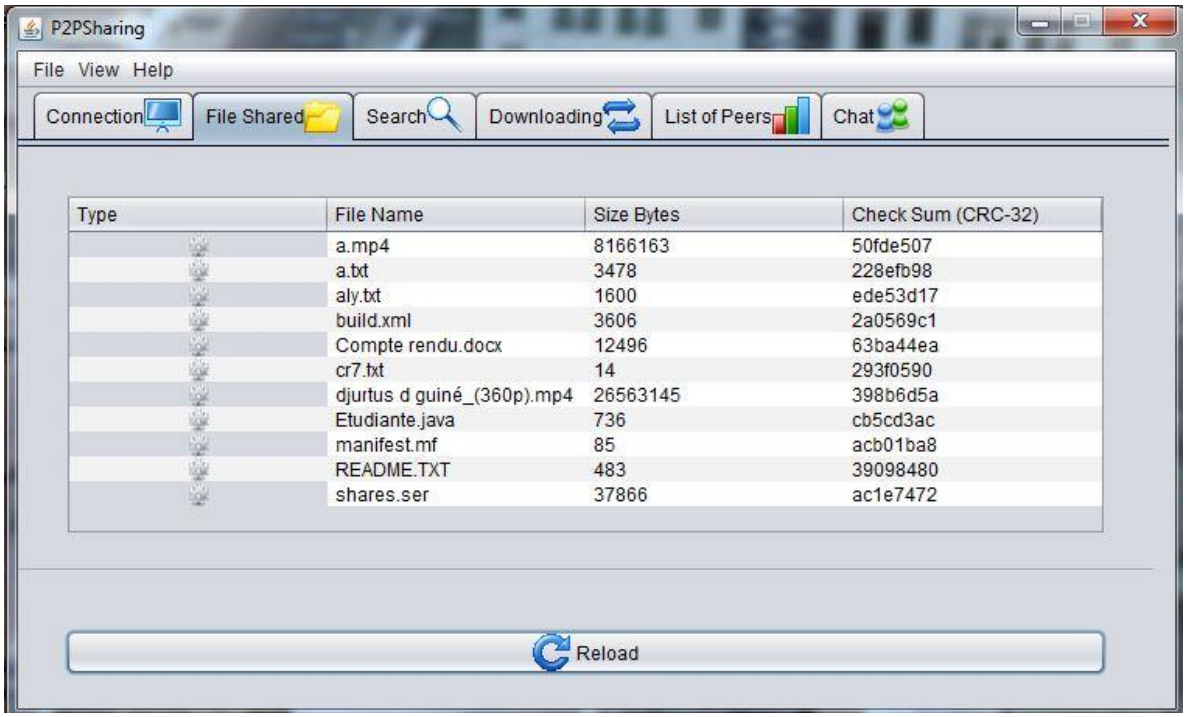


Figure 3.11:l'onglet partage du dossier avec la liste de fichiers partagés

Recherche de fichier

Cet onglet a été conçu de telle sorte que les pairs peuvent rechercher un fichier spécifique dans le groupe. La demande de contenu sera ensuite envoyée à tous les pairs disponibles dans le groupe. Si un poste a demandé le contenu, les pairs qui ont ce fichier vont lui répondre avec les détails du contenu, afin qu'ils puissent être téléchargés comme montré dans la figure.

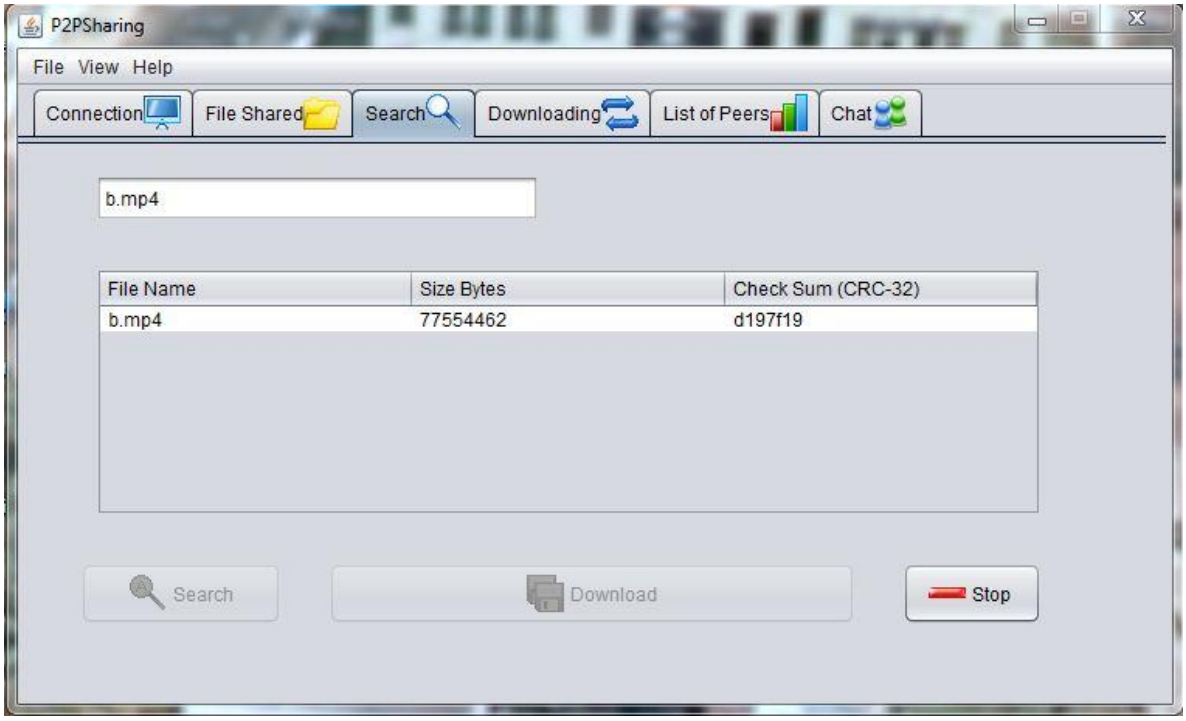


Figure 3.12: L'onglet recherche du fichier

Comme on peut le voir sur cette figure, il existe une colonne dans la liste des fichiers qui contient la somme de contrôle (checksum) - cette somme de contrôle a été reçue du pair qui fournit le fichier. Ces checksum nous assure que le téléchargement a réussi et que le fichier n'a pas été corrompu ce qui assure l'intégrité des fichiers téléchargés.

Lorsque l'utilisateur entre un nom de fichier, cela va commencer à rechercher un fichier au sein du groupe, et le progrès de recherche sera poursuivi jusqu'à ce que l'utilisateur arrête la recherche.

Téléchargement du fichier

Cet onglet présente toutes les informations concernant le processus de téléchargement en cours. Il comprend des informations telles que: Téléchargement de fichier, Statut du téléchargement, le pourcentage de téléchargement. Mais aussi, les utilisateurs peuvent interrompre leur processus de téléchargement par le bouton d'annulation. Le bouton de

vérification est également conçu pour vérifier le fichier une fois le téléchargement est terminé pour assurer que le téléchargement n'a pas été corrompu. La figure suivante donne un aperçu de l'onglet Téléchargement.

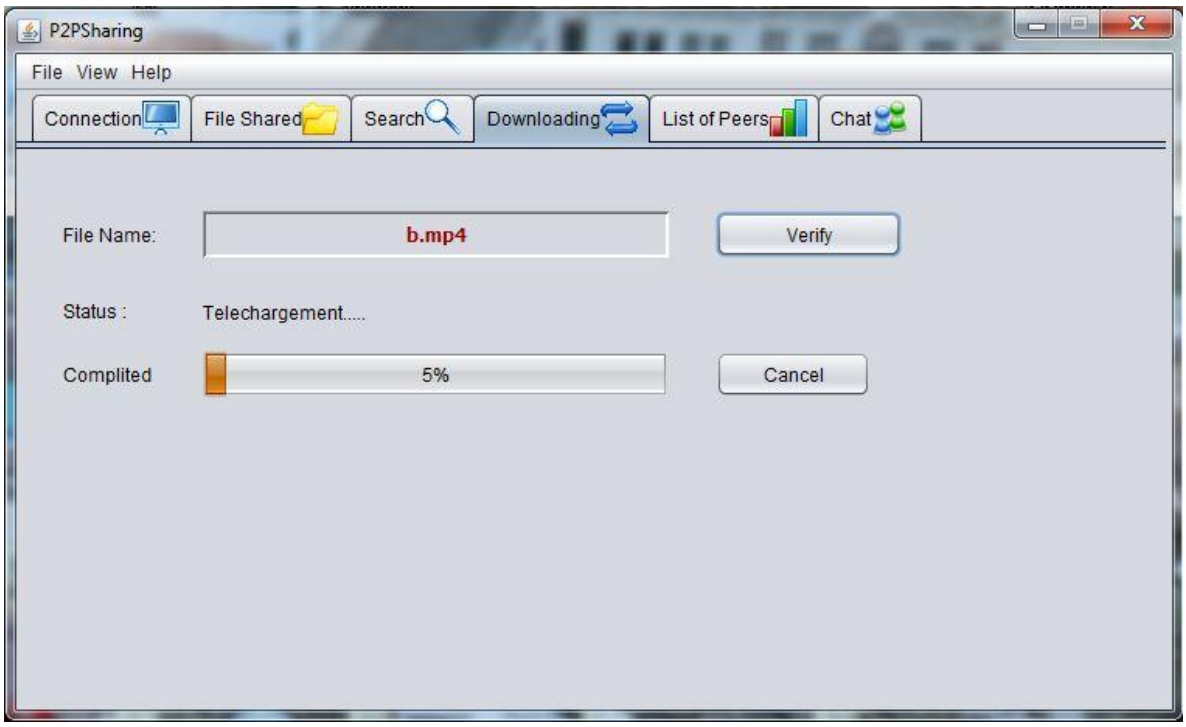


Figure 3.13:l'onglet téléchargement du fichier

Liste des paires

Ce panneau tente de trouver les pairs qui font partie du même groupe et qui sont connectés, une fois le bouton « trouver » cliqué la recherche se lancera et arrêtera une fois le bouton stop est cliqué. La figure suivante donne un aperçu de cet onglet.

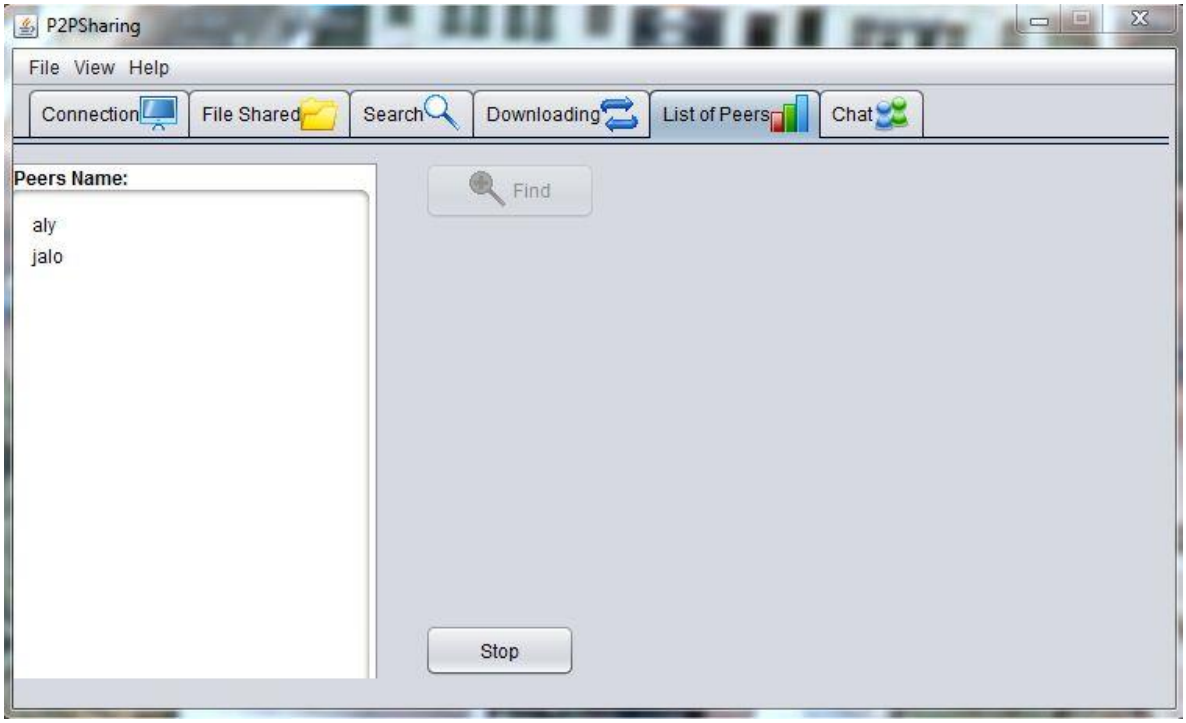


Figure 3.14: l'onglet Liste des pairs connectés

Chat

Dans cet onglet il a la possibilité de chatter. Toutes les paires connectées, peuvent faire des échanges des messages. Les messages seront vus pour tous les membres connectés.

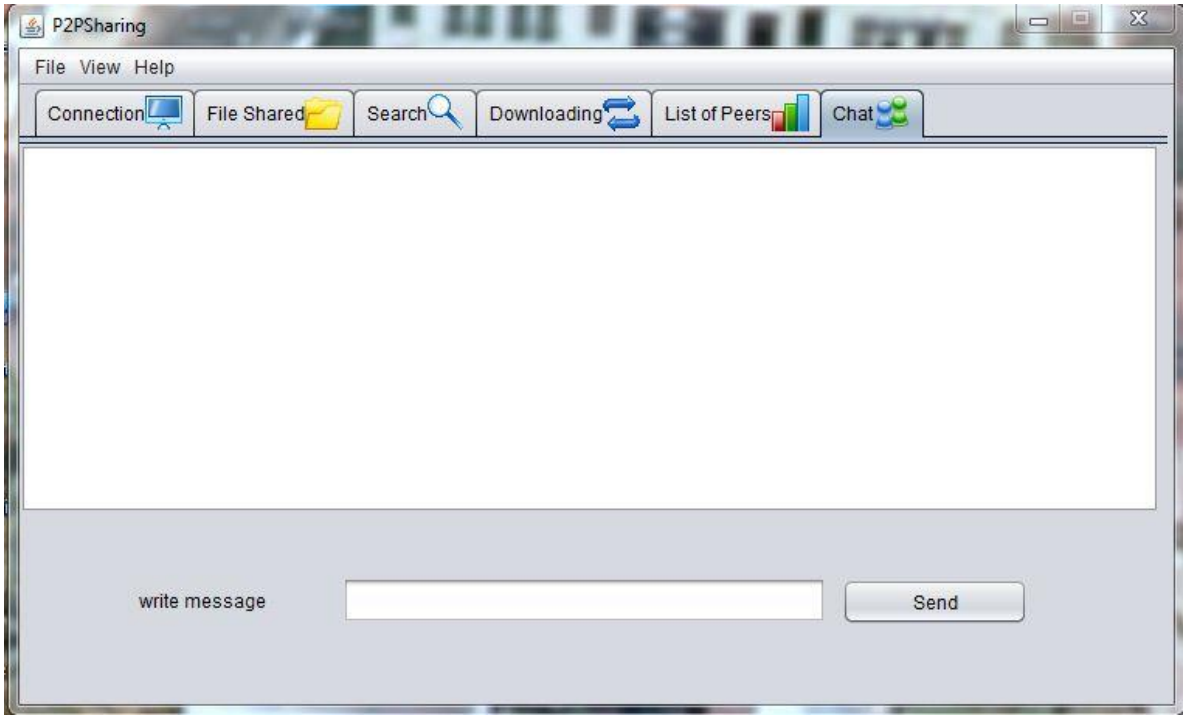


Figure 3.15: L'onglet chat

3.7. Conclusion :

Dans ce chapitre on a parlé de la réalisation de notre application, et comment elle a été réalisée. Les entités de la bibliothèque jxta utilisé ont été présentées. J'ai fait quelques captures d'écran en guise de démonstration de l'interface de cette application ainsi que les autres interfaces telles que le résultat lors des recherches des fichiers. L'application est basée sur jxta et développé avec java, et donne aux utilisateurs la possibilité de travailler dans la langue qu'ils souhaitent (anglais, français ou arabe).

Conclusion Générale

Ce projet est une étude basée sur le P2P, en ce qui concerne l'application du partage de fichiers, afin donner un aperçu de la technologie P2P. L'aspect principal de ce projet a été de créer une application de partage de fichiers P2P décentralisé utilisant la technologie JXTA.

Cette application est complètement décentralisée, ce qui veut dire qu'aucun serveur d'indexation ou de routes du réseau n'est utilisé, nous décidons d'utiliser cette technologie car elle est en plein essor et nous croyons que c'est l'avenir des systèmes distribués. Ce projet présente un aperçu des systèmes p2p qui pourra aider les futurs étudiants à avoir un peu des connaissances et connaitre l'importance du p2p dans le monde actuel.

Pour bien réaliser cette application de fin d'études représentant le partage de fichiers dans les réseaux p2p qui permet le téléchargement de fichiers tel qu'ils soient et de partager aussi sans dépendre d'un serveur, j'ai utilisé la plateforme JXTA et l'application est écrite en langage de programmation java. Ce projet contient la partie théorique où j'ai présenté différents systèmes distribués, leurs avantages et inconvénients, j'ai aussi parlé des différents modèles du p2p et aussi les parties pratiques. Pour la réalisation de la partie pratique je me suis confronté à de nombreux problèmes au fur et à mesure que j'avançais. L'un de ces problèmes est la documentation de jxta qui est en très grande majorité en anglais que je ne maitrise pas du tout mais en plus elle est très rare. Ce qui m'a couté beaucoup de temps en plus je n'ai pas trouvé des exemples pour pouvoir m'inspiré rendant le projet encore plus difficile. Une autre contrainte de taille était la programmation avec jxta car on doit respecter des règles dont on ne peut pas s'en passer parce qu'on est obligé d'utiliser la base de jxta ce qui réduit plus ou moins ma liberté. Mais surmonter tous ces problèmes m'a rendu fier et je suis très heureux d'avoir pu achever ce projet. J'ai aimé cette aventure qui a été vraiment bénéfique. Je crois que j'ai accomplie l'objectif cependant il y a des améliorations qu'on peut apporter à cette réalisation et qui restent comme perspective pour ce travail. Parmi ces perspectives on peut citer l'amélioration de la vitesse de

téléchargement et le téléchargement de plusieurs fichiers en même temps; les utilisateurs doivent être en mesure de télécharger le même fichier à partir de différentes sources. Cela peut être fait en utilisant des bibliothèques supplémentaires pour JXTA.

Bibliographie

[1] Eric Cariou, «Introduction aux System Distribué», Université de Pau
[2] Tarak Chaari, « Cours System Distribué», Institut Supérieure d'électronique et communication
[3] Mohamed Mosbah , « Modeles et Approches Formels des System Distribué» ,Université de Bordeaux
[4] Vitor Duarte , « SistemasDistribuidos», Universidade de lisboa, 2010
[5]BelkhoucheSouheyla, « etude et admnistration des systemes de supervision dans un reseau local», université du tlemcen,2011
[6]Raymond poincare, « Cours Clients-Serveurs»
[7]Renato Fileto, « Curso de sistema de informação » ufscuniversidade de brasil,2006
[8]Patrick Marlier, « Securité du Peer to Peer», 2007
[9]Androutsellis. S- Theotokis and D. Spinellis. « A survey of peer to peer content distribuition technologies », 2004
[10]Radad Al King, « localistion des source des donné et optimisation de requetes repartie en environnement pair à pair», these de doctorat, université de toulouse,2010
[11]OtmaneBouhamida, « Modelisation et simulation du problem du trou noir dans les reseauxmobiles»these master de master , université de ouargala, 2013
[12]HoudaHafi, « Protocol pour la securité des reseaux sans fil peer to peer », these de magister de UKM Ouargla,2010
[13]Anne Benoit, « Cours Reseau Pair à Pair» université de lyon, 2006
[14]Antony Rowstron, «Pastry: Scalable, decentralizedobject location and routing for large-scale peer-to-peer systems» Rice University
[15]Belhocine, « reseau Pair a Pair Structuré», Cours M2 RSD, departementd'informatiqueuniversité de tlemcen, 2016
[16]EkaterinaChtcherbina, « Project JXTA- Guide to peer –to-peer framework» university of Munich
[17]Joseph D Gradecki, « Mastering JXTA : Building Java Peer -to-Peer Application»,Books Packaging

Résumé

Le réseau pair à pair est en nette évolution dans ces dernières années et ces avantages ont pu surmonter de nombreux problèmes qui ne peuvent pas être résolue avec l'architecture classique clients/serveurs ou bien ça sera très difficile à réaliser et très couteux.

Ces nombreux handicaps dans l'architecture clients/serveurs, nous a poussé à développer dans un premier temps une application de partage de fichiers pleinement fonctionnelle en peer-to-peer totalement décentralisé et qui assure une recherche déterministe, qui gère le dynamisme au sein du réseau tout en facilitant son utilisation via une interface graphique multi langage, puis nous l'avons amélioré en donnant l'opportunité aux utilisateurs de se communiquer entre eux via un forum de chat.

Mot clé. Pair á pair, décentralisé, client/serveur, application, partage, chat.

Abstract

Recently peer to peer architectures have seen lot of progression due to their benefits and how they can overcome many problems that can not be solved with classical architecture such us client / server architecture, or it will be very difficult and very expensive in the other case.

In order to overstep the client / server architecture limits, we proposed a fully functional file sharing application completely decentralized which provides a deterministic search, manages with the dynamism within the network while facilitating its use via a rich and a multi language graphical interface, after that we have improved our application by giving the opportunity for our users to communicate with each other via a chat forum.

Key words : peer to peer, décentralized, client/server, application, sharing, chat.

ملخص

تشهد شبكة الند للند تطورا كبيرا في السنوات الأخيرة ،حيث استطاعت التغلب على العديد من المشاكل التي تعاني منها الانظمة المركزية التي تعتمد على جهاز خادم التي نقترح حلولا صعبة الإنشاء مكلفة جدا في مثل هذه الحالات.

من هذا المنطلق قمنا بإنجاز تطبيق يمكن من تبادل الملفات من خلال شبكة ند للند ترتكز على اللامركزية كما انها تضمن دقة في البحث وتسير بطريقة جيدة الحركية الكبيرة التي قد تحدث في الشبكة. يتم استعمال لتطبيق من خلال واجهة متميزة ومتعددة اللغات. ثم قمنا بعد ذلك بإضافة بعض التحسينات من خلال تمكين المستخدمين من التواصل في ما بيهم عن طريق منتد ى للدردشة.

الكلمات المفتاحية: الند للند ،اللامركزية ،انظمة مركزية ،تطبيق ،تبادل ،دردشة