

## ***ÉTUDE ET ANALYSE DES RÉSEAUX PAIR-À-PAIR***

*Ce chapitre présente et analyse les réseaux pair-à-pair (P2P) qui sont des réseaux de recouvrement spécifiques. Il élucide leurs différentes architectures et leurs principaux protocoles. Il aborde plus particulièrement ceux à base des tables de hachage distribuées (DHT).*

### **1.1 - Réseaux de recouvrement**

Ce paragraphe définit les réseaux de recouvrement et en donne quelques exemples

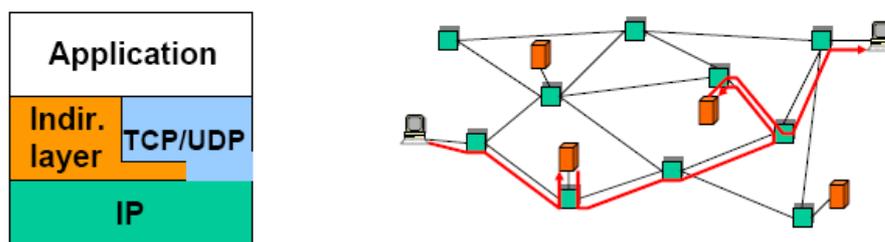
#### ***1.1.1 - Définition***

Un réseau de recouvrement ou *overlay* est un réseau virtuel s'appuyant sur un ou plusieurs réseaux physiques existants. Parmi ces réseaux existants, dits sous-jacents ou *underlay*, l'Internet (ou des sous-réseaux de l'Internet) est un bon exemple.

Dans la suite, ce sont les qualificatifs anglais : « *overlay* » pour un réseau de recouvrement et « *underlay* » pour un réseau sous-jacent, qui seront privilégiés.

Le réseau *overlay* est formé d'un sous-ensemble des nœuds du réseau *underlay* et d'un ensemble de liens logiques entre eux. Ces liens permettent une communication « directe » entre les nœuds concernés, en faisant abstraction de la topologie et des protocoles du réseau *underlay*. Nous désignons par nœud tout équipement du réseau, terminal ou intermédiaire, de quelque nature qu'il soit, jouant le rôle de routeur, client ou serveur.

Les réseaux *overlays* ont vu le jour dès lors qu'un nouveau service, inexistant dans le réseau, devait y être implanté. Ainsi tout problème informatique peut se résoudre par une couche d'indirection ou plutôt une redirection vers un nouveau réseau virtuel qui implémente la solution (cf. fig. 1.1). Aussi peut-on concevoir une superposition de réseaux *overlays*. Dans ce cas, les réseaux *overlays* supérieurs s'appuient sur un ou plusieurs autres réseaux intermédiaires ayant l'Internet pour réseau *underlay*.



**Figure 1.1** : Indirection et réseau *overlay*<sup>1</sup>

### 1.1.2 - Quelques exemples

Voici quelques exemples de réseaux *overlays* : Resilient Overlay Network (RON) [ABK<sup>+</sup>01], Content Delivery Network (CDN) [Mor03] et Eternity [And96].

#### 1.2.4.1 - Resilient Overlay Network

Resilient Overlay Network (RON) [ABK<sup>+</sup>01] est une architecture *overlay*, basée sur l'Internet. Elle surveille la qualité des liens Internet pour permettre la détection rapide de routes alternatives lors des interruptions de service des applications unicast distribuées. Elle assure ainsi le rétablissement d'une communication entre un groupe de nœuds en quelques dizaines de secondes au lieu de plusieurs minutes, comme c'est le cas avec BGP-4 [RFC1771]. Il a été, en effet, observé qu'il est parfois plus rapide d'envoyer

<sup>1</sup> **In** : [Sto04]

un message via un ou plusieurs intermédiaires, que de l'envoyer directement à sa destination.

#### 1.2.4.2 - Content Delivery Network

Content Delivery Network (CDN) [Mor03] est un réseau de serveurs cache offrant un dispositif de gestion optimisée des flux de données grand consommateurs de bande passante, permettant leur transit dans de bonnes conditions et empêchant les goulots d'étranglements.

Les clients du CDN sont des fournisseurs d'accès au réseau ou des fournisseurs de contenu qui souhaitent fiabiliser leur disponibilité. Les principaux fournisseurs sont Akamai [Aka] et Cisco [Cis].

Il est à noter qu'un réseau de diffusion de contenu ne se limite pas au streaming (diffusion en temps réel de contenus audio/vidéo). Aussi est-il possible qu'un réseau pair-à-pair soit utilisé en tant que réseau de diffusion de contenu, moyennant les outils logiciels adéquats.

#### 1.2.4.3 - Eternity

Eternity [And96] a été développé dans le but de construire un moyen de stockage résistant aux attaques par déni de service. C'est un modèle pair-à-pair de stockage anonyme qui assure la réplication des données et la redondance des chemins. Il n'a jamais été implémenté.

## 1.2 - Caractéristiques et pratiques des réseaux pair-à-pair

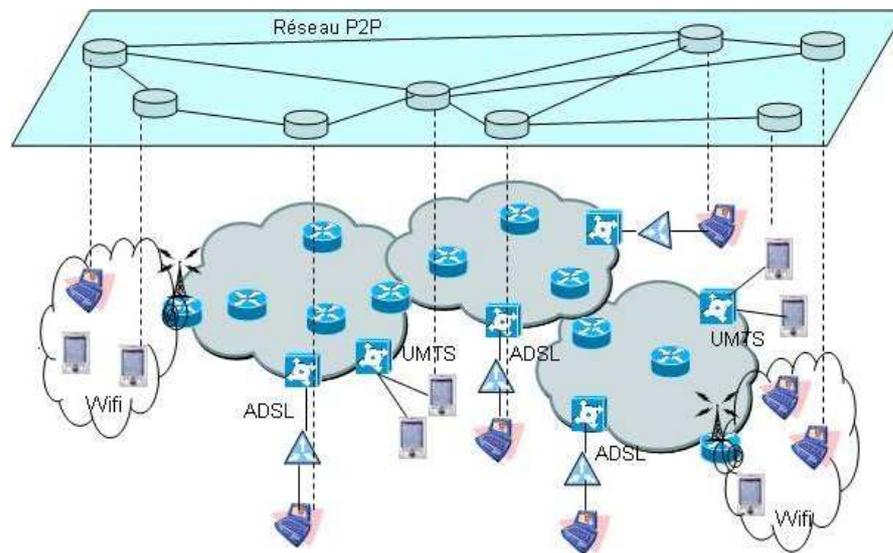
### 1.2.1 - Définition

Le P2P est une contraction de *Peer to Peer*, en anglais, ou, *Pair à Pair*, en français. Le terme P2P sera utilisé par la suite.

Les définitions du P2P sont nombreuses et différentes. La plus générique est la suivante.

Le P2P fait référence à une classe de systèmes et d'applications qui utilisent des ressources distribuées (d'un réseau) pour réaliser une fonction critique de manière décentralisée [MKL<sup>+</sup>02]<sup>1</sup>. Il est à noter que le terme fonction fait allusion à un service, un mécanisme (par exemple, de routage,...), etc.

La figure 1.2 illustre un modèle de réseau P2P.

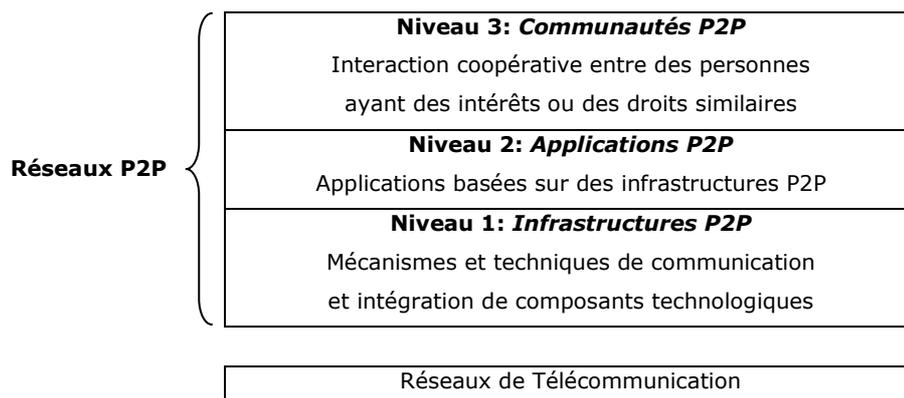


**Figure 1.2 : Modèle de réseau P2P**

### 1.2.2 - Différentes approches

Dans la multitude des définitions des réseaux P2P, nous distinguons différentes approches qui justifient cette diversité. Trois aspects des réseaux P2P peuvent être identifiés, superposés en niveaux distincts au dessus d'un réseau de communication. Ces niveaux sont – de bas en haut : les infrastructures, les applications et les communautés (cf. fig. 1.3) [SF03].

<sup>1</sup> "P2P refers to a class of systems and applications that employ distributed resources to perform critical function in a decentralized manner."



**Figure 1.3 :** Différents niveaux d'un réseau P2P

#### 1.2.2.1 - Infrastructures pair-à-pair

Les infrastructures P2P se positionnent au dessus de réseaux de communication existants, qui constituent alors la base de tous les niveaux P2P. Elles définissent la topologie et le routage et sont parfois gérées par des algorithmes spécifiques. Elles fournissent des mécanismes et techniques de communication et d'intégration des composants IT, et parfois même des mécanismes d'authentification pour une éventuelle sécurisation des services P2P offerts.

#### 1.2.2.2 - Applications pair-à-pair

Les applications P2P constituent un niveau intermédiaire entre d'une part les communautés P2P, qui les utilisent, et d'autre part les infrastructures P2P dont elles utilisent les services. C'est à ce niveau que peuvent être gérées les différentes ressources : fichiers, bande passante, capacité de stockage, etc.

En permettant l'échange entre des communautés, les applications génèrent le trafic de données utiles recherchées, mais aussi un trafic – appelé par la suite « de signalisation » – formé par les requêtes et les différentes mises à jour.

À ce niveau, tout comme au niveau des infrastructures, le terme *peer*, ou pair, représente tout composant IT du réseau P2P.

#### 1.2.2.3 - Communautés pair-à-pair

Les communautés P2P émergent de l'interaction coopérative entre des personnes ayant des intérêts ou des droits similaires.

Elles représentent le phénomène apparent au niveau social de l'émergence et du développement des réseaux P2P. Du coup, à ce niveau, le terme *peer*, ou pair, prend un sens non technique et représente toute personne utilisant un service P2P à travers des logiciels ou autres applications P2P.

#### 1.2.3 - *Caractéristiques générales*

Un réseau P2P est un réseau logique distribué dont le principe de base est un principe d'égalité entre des nœuds dynamiques. Ces nœuds sont donc dits *pairs* bien que, comme nous le verrons dans la suite, ce principe ne soit pas vérifié dans tous les types d'architecture P2P.

Les pairs peuvent être de nature différente (routeurs ou périphériques autonomes quelconques), mais sont fonctionnellement identiques, pouvant jouer à la fois le rôle de client, serveur ou routeur.

Le routage se fait au niveau applicatif, dans le sens où il est indépendant du routage dans le réseau sous-jacent – qui est généralement un routage IP. Le routage P2P est généralement basé sur l'usage de clés et consiste en un échange direct entre les pairs. Il utilise les ressources des différents pairs, éliminant ainsi les coûts d'infrastructure.

Chaque pair participant à un réseau P2P met à disposition une partie ou la totalité de ses ressources (données, bande passante, espace de stockage, mémoire cache, CPU, etc.), mais aussi des services (impression, etc.). Chaque pair n'a qu'une visibilité partielle du réseau.

Cette topologie virtuelle – qu'est un réseau P2P – peut avoir une distribution géographique aussi importante que celle du réseau Internet. Elle est aussi instable, vu que la durée de présence de chaque pair est assez différente et variable.

### 1.3 - Architectures types des réseaux pair-à-pair

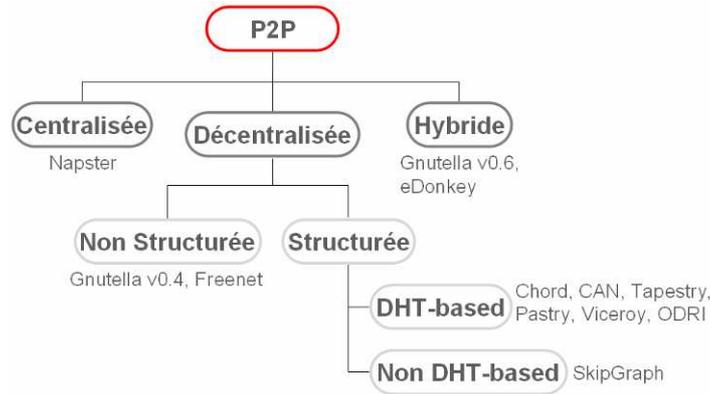
Plusieurs modèles d'architecture P2P existent. D'ordinaire, c'est le degré de décentralisation qui est retenu pour leur classification, puisque c'est là l'idée de base du P2P. Ce degré de décentralisation de l'architecture caractérise aussi l'index de référence des ressources.

La première génération des applications P2P était à architecture centralisée. Une deuxième génération à architecture décentralisée non structurée a vite vu le jour, suivie par une troisième génération d'applications à architecture hybride.

Au niveau de l'infrastructure P2P, des protocoles et des architectures décentralisées structurées se sont développées en parallèle aux applications P2P, à partir de la deuxième génération.

Nous distinguons donc trois familles d'architectures P2P (cf. fig. 1.4) :

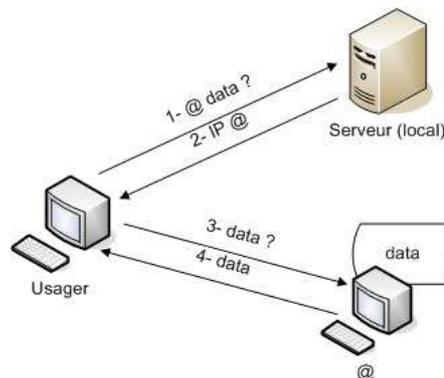
- l'architecture centralisée ;
- l'architecture décentralisée, qui peut être structurée ou non structurée ;
- et l'architecture hybride.



**Figure 1.4 :** Classification des systèmes P2P, avec quelques exemples

### 1.3.1 - Architecture centralisée

L'architecture centralisée est caractérisée par un index central pour repérer les ressources. Cet index reçoit tous les messages de contrôle et toutes les requêtes, mais par la suite, l'échange se fait directement entre les pairs concernés (cf. fig. 1.5). C'est ce qui distingue principalement cette architecture d'une architecture traditionnelle de type client-serveur.



**Figure 1.5 :** Architecture P2P centralisée

Comme toute architecture centralisée, cette architecture facilite la gestion des différentes ressources de l'index et offre une bonne performance de découverte des ressources (le coût de localisation des ressources est faible). Cependant la centralisation représente un goulot d'étranglement au niveau de l'index, tant par la charge de son

utilisation que de sa mise à jour. De plus, au niveau de la sécurité, l'index est le talon d'Achille de ce type d'architecture.

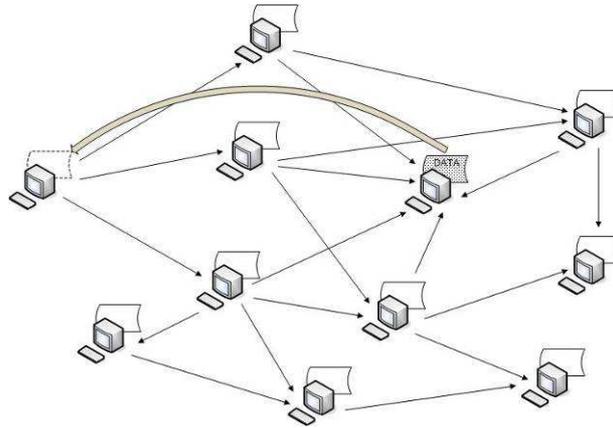
L'exemple type d'une telle architecture est Napster (1999-2002) [Nap99] qui a déclenché le *phénomène* P2P. Il a aussi été le modèle le plus spectaculaire de la réussite technologique du P2P. Avec Napster, les pairs se joignant au réseau informaient l'index des fichiers dont ils disposent, et le recontactaient pour obtenir les coordonnées (adresse IP et numéro de port) d'un autre pair possédant les fichiers recherchés.

BitTorrent [Bit, CC03, Cha05] est un autre exemple d'architecture P2P, qui était cité dans le cadre des architectures partiellement centralisées, à cause de ses différents *trackers* (sorte d'index central pour la localisation d'une même ressource). Depuis l'introduction d'une couche réseau basée sur le protocole Kademlia [MM02] (cf. p. 54), qui a pour effet de distribuer les informations de chaque *tracker*, BitTorrent est déjà officiellement décentralisé [Cha05, Loe08].

### **1.3.2 - Architecture décentralisée**

#### **1.3.2.1 - Caractéristiques**

Dans l'architecture P2P décentralisée, tous les pairs sont fonctionnellement parfaitement équivalents. C'est un modèle P2P pur. (cf. fig. 1.6) Cette architecture est caractérisée par une décentralisation de l'index, qui devient local à chaque pair, faisant effet de table de routage. La décentralisation rend le système autonome et répartit la charge équitablement. L'architecture décentralisée est donc plus robuste que l'architecture centralisée, mais le temps de découverte des ressources est évidemment plus long.



**Figure 1.6 :** Architecture P2P décentralisée

On distingue dans ce type d'architecture deux sous-catégories :

- l'architecture décentralisée non structurée ;
- et l'architecture décentralisée structurée.

#### 1.3.2.2 - Architecture décentralisée non structurée

Une architecture P2P décentralisée est dite non structurée lorsqu'aucune contrainte topologique n'existe. Un système P2P à architecture décentralisée est aussi dit : système P2P pur. Aucune connaissance de la topologie n'est donc disponible au préalable et chaque pair dans le réseau est autonome. La maintenance et la mise à jour des connexions se font par sondage périodique du voisinage, et la découverte des ressources se fait par une technique d'inondation associant un champ TTL (Time To Live) à chaque message de recherche envoyé, afin de comptabiliser le nombre de retransmissions restantes.

L'architecture décentralisée et non structurée permet de pallier les points faibles de la centralisation. Elle est simple et flexible, et ne requiert pas des requêtes exactes ou des demandes de recherches très précises. Néanmoins, cette architecture ne peut pas être gérée (par un opérateur de réseau ou fournisseur de services). Elle présente aussi

certains inconvénients résultant de l'utilisation de la technique d'inondation et du TTL :

- la génération d'un nombre important de messages, dont découlent :
  - une consommation importante de la bande passante – et donc une consommation des ressources du réseau ;
  - la visite des pairs par des messages (requêtes ou réponses) non sollicités ;
- l'absence de garantie de résultat et de connectivité ;
- un passage à l'échelle assez limité.

L'exemple type d'une architecture P2P décentralisée non structurée est Gnutella dans sa version initiale [Cli01, Rip01]. Le TTL y était typiquement fixé à 7. De ce fait, une mise hors réseau (panne ou déconnexion) d'un nombre aléatoire de pairs scinderait le réseau en deux.

FreeNet [CSW<sup>+</sup>00] est un autre exemple qui utilise un routage par inondation. Mais il s'agit déjà d'un pont vers les architectures décentralisées structurées. En effet, les données qui y sont stockées ont déjà un identifiant : une clé. De plus d'autres types de clés sont aussi utilisés, comme pour la signature des documents et leur vérification. FreeNet assure l'anonymat et la permanence des informations qui y résident, en insérant chaque clé sur tout chemin parcouru. Cependant, ce fonctionnement présente des risques de sécurité, tels une attaque de type *man-in-the-middle* ou de type *cheval de Troie*.

### 1.3.2.3 - Architecture décentralisée structurée

Une architecture P2P décentralisée est dite structurée lorsque la topologie logique est contrôlée par un algorithme et les données sont mises en réseau aussi suivant un

algorithme bien spécifié. Ceci permet à n'importe quel pair de se joindre au réseau ou de le quitter, sans pour autant mettre en cause la cohérence de ce dernier ; un tel réseau est dit « auto-organisé ».

Un autre avantage d'une architecture décentralisée structurée est la garantie qu'elle offre sur les résultats des requêtes.

Ce type d'architecture sera détaillé à la section 1.4 (cf. p. 40), qui lui est consacrée et où différents exemples seront présentés.

### ***1.3.3 - Architecture hybride***

L'architecture hybride est une architecture décentralisée dont chaque nœud est l'élément central d'une architecture centralisée. Ces éléments centraux sont des super-pairs par rapport à l'architecture globale.

Pour une homogénéité entre les super-pairs, c'est souvent le critère de la bande passante – identifiée suivant le type de connexion (e.g. par modem ou haut débit) – qui est pris en compte. Mais d'autres critères peuvent aussi entrer en jeu, comme la puissance de calcul, l'espace de stockage, etc. Un nœud peut, avec le temps, à plusieurs reprises gagner et perdre le statut de super-pair.

L'architecture hybride tire avantage simultanément de l'architecture centralisée et de l'architecture décentralisée. Le nombre de pairs mis en jeu dans la découverte des ressources est aussi réduit, et avec lui, le trafic global entre les pairs. Ceci permet une économie de bande passante et facilite le passage à l'échelle.

Un super-pair a cependant les mêmes faiblesses que l'index d'une architecture P2P centralisée. Une amélioration possible est d'assurer une redondance en choisissant, pour un même sous-groupe, un ou plusieurs partenaires au super-pair [YGM03]. L'architecture sera dite à super-pairs redondants.

Des supers-pairs partenaires ont les mêmes connexions aux autres pairs et possèdent des index identiques de toutes leurs ressources. Pour une répartition de charge équitable entre les super-pairs partenaires, les pairs d'un sous-groupe envoient leurs requêtes selon le principe d'ordonnement *round Robin*.

L'introduction dans l'architecture P2P hybride de super-pairs partenaires améliore la tolérance aux fautes. Et, avec  $k$  partenaires, la charge d'un partenaire est divisée par  $k$ . En revanche, le coût d'adhésion d'un nouveau pair au réseau est multiplié par  $k$  et le nombre de connexions ouvertes par  $k^2$ .

Voici quelques exemples d'applications basées sur un réseau P2P hybride : Gnutella (à partir de la version 0.6) [SR04], FastTrack [Har04], eDonkey/eMule [eMu, KB05], Skype [Sky, BS06a, GDJ06].

Il est à noter que les architectures hybrides et les architectures centralisées sont parfois regroupées au sein d'une même classe dite architecture à serveurs.

#### **1.3.4 - JXTA**

JXTA [JXTA, Dal03] abréviation de "JuXTApose", par rapport au modèle de référence (Web ou client-serveur), est une plateforme associée à une suite de protocoles libres de droit. Elle a été développée par SUN Microsystems dans le but de supprimer les problèmes de non-interopérabilité et non-portabilité des différentes implémentations d'architectures P2P. Ainsi JXTA permet d'interconnecter n'importe quel système sur n'importe quel réseau, établissant des communications et des échanges au sein de réseaux fixes et mobiles, filaires et non filaires (PC, PDA, etc.), même en présence d'un pare-feu. Par conséquent, l'on a plusieurs points d'accès au réseau (IP, TCP, HTTP, etc.) et les liens sont établis dynamiquement. Les canaux de communications virtuels ainsi formés, et appelés « *pipe* », fonctionnent en mode asynchrone.