



HAL
open science

Allocation optimale de ressources dans des overlays

Gwendal Simon

► **To cite this version:**

Gwendal Simon. Allocation optimale de ressources dans des overlays. Evolution des technologies pair-à-pair : optimisation, sécurité et application, Hermès Science, Lavoisier, 2010, 978-2-7462-2579-4. hal-02367725

HAL Id: hal-02367725

<https://hal.science/hal-02367725v1>

Submitted on 18 Nov 2019

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

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

Allocation optimale de ressources dans des overlays

Gwendal SIMON

30 novembre 2009

Chapitre 1

Allocation optimale de ressources dans des overlays

1.1. Introduction

En sciences économiques, une ressource est considérée comme *rivale* quand son allocation à un utilisateur rend impossible son allocation à un autre utilisateur. Cette notion de rivalité est au cœur de l'histoire tourmentée des réseaux pair-à-pair. Tout d'abord, les systèmes de téléchargement pair-à-pair ont permis à des millions d'utilisateurs de s'échanger, directement, des fichiers en profitant du fait que les fichiers numériques sont par essence des biens non-rivaux. En effet, il est possible de copier un fichier infiniment, donc un utilisateur peut conserver un fichier et le donner à d'autres utilisateurs. Au-delà de ces échanges de biens non-rivaux, les architectures d'échanges pair-à-pair se sont développées avec une promesse : la conception de systèmes capables de fournir un service à un nombre illimité d'utilisateurs. En effet, les systèmes classiques nécessitent que le fournisseur de service consomme des ressources rivales : les ressources de stockage, les ressources de calcul ou les ressources de bande-passante. Là encore, les réseaux pair-à-pair ont modifié significativement un contexte établi. Quand un serveur consomme des ressources pour servir individuellement chaque client, il admet naturellement une limitation dans le nombre de clients qu'il peut satisfaire. Au contraire, si un client met immédiatement à disposition d'autres clients ses propres ressources, c'est-à-dire s'il devient serveur pour d'autres clients, alors la quantité de ressources disponibles augmente proportionnellement au nombre de clients. Ainsi, les ressources rivales en quantité suffisante peuvent soutenir des échanges de biens non-rivaux qui se multiplient. La réalité s'est pourtant avérée plus complexe.

Tout d'abord, l'utilisation massive des applications pair-à-pair a rapidement révélé que de nombreux utilisateurs adoptaient un comportement égoïste. En refusant de mettre à disposition des autres utilisateurs leurs ressources, ces utilisateurs égoïstes ont semblé briser la proportionnalité établie entre quantité de ressources mises à disposition et nombre d'utilisateurs. Finalement, de nombreux travaux basés sur des analyses d'applications réelles ont montré qu'un pourcentage d'utilisateurs généreux relativement faible pouvait suffire à assurer un service dont la qualité paraît satisfaisante aux utilisateurs [VAS 09]. D'un point de vue pratique, ces comportements égoïstes ont eu pour principal effet d'allonger substantiellement le temps nécessaire au téléchargement d'un fichier. En effet, si un fichier est un bien non-rival, sa transmission d'un pair vers un autre pair nécessite d'utiliser de la bande-passante, une ressource rivale d'autant plus rare que la quantité de bande-passante disponible augmente plus lentement que le nombre de clients en cas de comportements égoïstes. Une abondante littérature scientifique s'est intéressée à la conception de mécanismes d'incitation permettant la réduction des comportements égoïstes. Le système *bit-torrent*, dans lequel la quantité de ressources rivales mis à disposition des autres pairs conditionne la qualité du service, a provisoirement résolu ce problème d'égoïsme pour les applications d'échange de fichiers [QIU 04].

D'autres problèmes sont apparus récemment. Le premier est inhérent à l'architecture basée sur un overlay. Un participant à un système pair-à-pair de très grande taille ne peut pas raisonnablement être connecté à tous les autres participants. Il est en relation directe avec un sous-ensemble, relativement faible, d'autres participants, qu'on appelle ses voisins. L'allocation de ressources rivales, notamment la bande-passante, s'établit entre voisins. Chaque pair peut être caractérisé par sa capacité, c'est-à-dire la quantité de ressources rivales qu'il est capable de mettre à disposition de ses voisins, et par sa demande, la quantité de ressources rivales qu'il souhaiterait que ses voisins lui allouent. Les relations logiques entre voisins peuvent conduire à des situations où un pair ayant une grande capacité est incapable de l'allouer totalement parce que la demande de ses voisins est satisfaite. C'est le cas par exemple dans un système pair-à-pair de diffusion d'un flux. La demande des pairs est alors la quantité de données par seconde du flux. Si un pair a une capacité permettant de transmettre k fois ce flux, mais s'il est relié dans l'overlay à un nombre inférieur à k voisins, sa capacité sera fatalement sous-utilisée. Cela est d'autant plus paradoxal que, parallèlement, des pairs peuvent être sous-provisionnés à d'autres endroits de l'overlay. Tant que les overlays pouvaient être facilement modifiés (par exemple en fermant quelques connexions avec des voisins et en ouvrant de nouvelles connexions avec d'autres voisins), ce potentiel mauvais équilibre initial des ressources n'était pas un problème très grave. Mais, de plus en plus, des paramètres externes contraignent la topologie de l'overlay. Ainsi, la proximité réseau entre les nœuds est devenue un principe fondamental dans l'ouverture des connexions entre les pairs. Dans un autre cas, l'overlay a plusieurs utilités, et l'une d'entre elles prime. Par exemple, dans un monde virtuel en pair-à-pair, l'overlay sert principalement aux échanges de messages notifiant des

événements, et, en supplément, au transfert de données multimedia auto-produits. Ainsi, les overlays ne sont pas toujours dynamiques et pas toujours optimisés pour une fonction. Dans ce contexte, un axe de recherche en émergence étudie la capacité des overlays [CUI 04, JIN 07, LIU 08, YAN 06].

Le second problème est lié aux biens non-rivaux qui sont échangés. Dans un système d'échange de fichiers statiques, la notion de demande n'est pas très explicite. Une faible allocation de ressources en bande-passante se traduit par un long temps de téléchargement, ce qui réduit la qualité de service, mais garantit cependant un accès au service. Dans les systèmes de transmission de flux de données, il existe une valeur minimale de bande-passante à allouer à un pair pour qu'il puisse jouir du service ; un client ne peut pas lire une vidéo de k octets par seconde s'il ne reçoit que k' octets par seconde de données, pour toute valeur k' strictement inférieure à k . Or les données échangées sont de taille toujours croissante, tandis que les capacités en ressources rivales (bande-passante montante notamment) ont eu tendance à augmenter moins vite (voire stagner) ces dernières années. Le risque que la quantité de capacités mises à disposition soit inférieure à la somme des demandes devient élevé. La gestion des ressources, dans un contexte de grande hétérogénéité, est alors cruciale, puisqu'il est hors-de-question de se satisfaire d'une allocation de ressources qui ne soit pas optimale.

Dans ce chapitre, nous décrirons les travaux de recherche récents qui ont été menés sur ce sujet de gestion des ressources dans un environnement pair-à-pair [LEB 08, CHA 09]. Ce sont essentiellement les deux principaux résultats théoriques sur lesquels la recherche scientifique va pouvoir se référer pour bâtir des algorithmes visant à optimiser le comportement des systèmes pair-à-pair. Pratiquement, il s'agit pour un pair de déterminer comment allouer ses ressources, limitées, à ses voisins de manière que, collectivement, l'allocation produite soit optimale.

Le modèle que nous allons utiliser place des contraintes sur les nœuds et néglige les contraintes sur les arcs qui relient ces nœuds. Ce modèle n'est pas traditionnel dans les réseaux où il est plus courant de considérer qu'un lien a une capacité limitée. Il a été introduit et argumenté dans la littérature scientifique dans [LIU 08, MAS 07].

Après avoir introduit ce modèle original, nous détaillerons deux problématiques. Dans un premier temps, nous nous intéresserons à l'optimisation de la gestion de ressources rivales limitées (par exemple la bande-passante) dans un cas spécial correspondant à ce qu'on appelle un *régime stationnaire*, c'est-à-dire dans lequel les biens non-rivaux (essentiellement les données) peuvent être totalement négligés. Dans ce contexte, un pair est toujours capable d'exploiter les ressources rivales qu'il a allouées à un autre pair, par exemple dans le cas de la bande-passante montante, il possède toujours une donnée à transmettre à ses voisins. Nous verrons que déterminer une allocation de ressources optimale est réalisable. Un opérateur de réseau pair-à-pair ayant une vision globale du système peut calculer la meilleure allocation en un temps

polynomial du nombre d'éléments. Par ailleurs, la solution se référant à un problème de flot bien connu, des algorithmes distribués peuvent être facilement adaptés pour la conception d'un protocole pair-à-pair permettant à un pair de déterminer son allocation de ressources.

Dans un second temps, nous intégrerons les biens non-rivaux dans le modèle, de manière à étudier des applications plus concrètes dans lesquelles les données sont émises par une source et doivent être envoyées massivement à l'ensemble des participants. C'est typiquement le cas d'un fournisseur de contenus ayant à diffuser massivement un ensemble de données. Nous nous pencherons plus particulièrement sur le cas où les données ne sont pas en excès, c'est-à-dire un gros fichier découpé en un nombre donné de sous-fichiers (*chunks*) ou un flux vidéo découpé en plusieurs descriptions indépendantes. Notre problème d'allocation de ressources devient alors un problème d'optimisation de la qualité de service perçue par les utilisateurs, la qualité de service étant une fonction du nombre de données distinctes reçues. Nous montrerons que, dans ce cas-là, le problème est NP-complet, c'est-à-dire que l'allocation optimale n'est pas calculable en un temps raisonnable. Nous présenterons également un modèle en programmation en nombres entiers permettant de calculer plus facilement des solutions optimales pour des petites instances. Ces deux résultats sont les deux principales avancées dans ce domaine.

1.2. Description du Modèle

Nous allons tâcher ici de décrire les différents éléments d'un système pair-à-pair dans un modèle où nous cherchons à analyser les échanges de ressources entre participants. Ce modèle a été récemment décrit dans [CHA 09].

1.2.1. Overlay

Un overlay peut être modélisé comme un graphe non-dirigé $G(V, E)$. Il consiste en un ensemble de nœuds V et un ensemble E des relations logiques entre eux. Dans ce modèle, un lien entre deux nœuds u et v dans le graphe signifie que ces deux nœuds se connaissent. Il est donc possible de réserver une quantité de ressources rivaux (par exemple la bande-passante) entre le nœud u et le nœud v et vice versa, puis d'utiliser cette réservation (par exemple pour envoyer des données). Le graphe G n'est pas nécessairement complet, même si de nombreuses études antérieures ont eu tendance à le considérer. Le graphe G correspond plutôt à une topologie pré-calculée ou une situation à un instant donné. Il serait possible d'étendre ce modèle à des contextes plus dynamiques, notamment en pondérant les arcs, mais, par souci de concision et de clarté, le modèle présenté ici se concentre sur une situation à un instant donné.

Un lien entre deux nœuds u et v dans E , noté $\{u, v\}$ peut supporter une allocation de ressources dans les deux directions, c'est-à-dire u peut allouer des ressources pour

v et v peut allouer des ressources pour u . En conséquence, chaque lien $\{u, v\}$ peut être transformé en deux arcs dirigés (u, v) et (v, u) . L'ensemble des arcs dirigés qui dérive de l'ensemble E est noté E^* .

1.2.2. Ressources rivales

Les ressources rivales qui sont au cœur de ce modèle sont typiquement la bande-passante (principalement montante) pour des applications comme les systèmes pair-à-pair de diffusion de flux vidéos, ou la capacité de stockage pour des applications de sauvegarde collaborative ou encore la capacité de calcul pour des applications de calculs en grille. Nous faisons l'hypothèse qu'une quantité de ressources rivales peut être divisée en plusieurs quantités de ressources, qui sont évidemment rivales également. Nous considérons donc qu'il existe une fonction \sum qui opère sur deux quantités de ressources et qui résulte en une quantité de ressources.

La quantité de ressources rivales qu'un pair u dans V est susceptible de fournir à d'autres pairs est noté $c(u)$. Chaque pair u est également associé à une demande, notée $d(u)$, qui correspond à la quantité de ressources qu'il espère se voir attribuer. Plus spécifiquement, la demande représente la quantité minimale de ressources rivales qui doivent être fournies à un pair pour que la qualité perçue du service soit suffisamment satisfaisante. La demande peut être un paramètre du système (par exemple la taille du flux dans un système de diffusion vidéo) ou peut dépendre de capacités physiques (par exemple la bande-passante descendante) ou des besoins individuels (par exemple le nombre de problèmes à résoudre dans une grille de calculs).

1.2.3. Allocation de ressources

Nous définissons maintenant formellement une allocation de ressources. La fonction w de E^* vers N permet d'associer à chaque arc dirigé de l'overlay une quantité de ressource. Par exemple, pour un arc $e = (u, v)$, le poids $w(e)$ peut correspondre à une quantité $w(e)$ de bande-passante montante réservée par u pour servir v .

Il serait trivial d'ajouter des contraintes sur les arcs pour se replacer dans un contexte plus conventionnel des études scientifiques en télécommunications, mais nous souhaitons préciser une nouvelle fois que la limitation du cœur du réseau paraît négligeable en comparaison des limitations que les pairs rencontrent dans leurs conditions d'accès au réseau.

1.2.4. Formulation de problèmes

Nous n'allons pas étudier de manière exhaustive tous les problèmes qui peuvent être dérivés du modèle défini ci-dessus. Nous allons plutôt porter notre attention sur

deux problèmes qui nous paraissent suffisamment génériques pour s'appliquer à un nombre étendu d'applications.

Le premier problème est sans doute le plus générique, nous l'appelons SRA (pour *Stationary Regime Allocation*). Dans ce problème, la présence de biens non-rivaux est négligée, de fait, il est admis que toutes les ressources rivales allouées d'un pair à un autre peuvent, à n'importe quel instant, être effectivement allouées. C'est typiquement le cas lors de transfert de très gros fichiers sous *bit-torrent* où, à partir de quelques minutes de participation, deux nœuds possèdent une très grande quantité de données *distinctes* qu'ils peuvent s'échanger librement. Cet état est appelé *régime stationnaire* dans [QIU 04]. Le problème d'allocation optimal de ressources peut alors s'exprimer comme suit.

PROBLÈME.– SRA

Étant donné un overlay $G(V, E^)$ et deux fonctions de distribution de capacité $c(u)$ et de demande $d(u)$ pour tout pair u dans V , il s'agit de déterminer une fonction d'allocation w de E^* vers N telle que la demande $d(u)$ de tout nœud u peut être satisfaite.*

Le second problème sur lequel nous allons nous concentrer apparaît plus couramment. Il prend en compte le fait que, parfois, un pair n'est pas en mesure d'utiliser les ressources qu'il a allouées à un autre pair. Dans ce cadre, nous revenons à des modèles un peu plus classiques de la recherche en réseau avec une source qui émet des biens non-rivaux qu'il s'agit de transmettre aux différents pairs. Comme nous allons le voir plus tard, le modèle conduit néanmoins à la définition d'un problème plutôt original. Nous allons appeler ce problème *K-DCDA* (pour *Data-Capacitated Distribution Arborescence*).

Tout d'abord, définissons le problème 1-DCDA. Nous considérons ici que les ressources rivales qui peuvent être allouées entre les pairs sont contingentes de la disponibilité d'une donnée pour le pair qui alloue ces ressources. La donnée peut ici être considérée comme un fichier, un flux, un morceau de flux ou un problème. En particulier, un nœud peut participer à la distribution de cette donnée si et seulement si sa propre demande a été satisfaite, c'est-à-dire si une ressource rivale a été allouée vers lui par un possesseur de la donnée. Le problème 1-DCDA peut alors être formulé ainsi. Étant donné un overlay $G(V, E^*)$, une source s et une fonction de distribution de la capacité $c(u)$ pour tout u dans V , il s'agit de déterminer une allocation de ressources w de E^* vers $\{0, 1\}$ de telle manière que le nombre de nœuds ayant un arc entrant non-nul soit maximisé. L'arborescence dont la racine est en s et formé par l'ensemble des arcs non-nuls doit en outre respecter que, pour chaque nœud u dans V , le nombre d'enfants de u dans l'arborescence doit être inférieur ou égal à $c(u)$, la capacité de u .

Il est maintenant possible de généraliser le problème 1-DCDA au problème *K-DCDA*. La généralisation s'applique sur le nombre de données indépendantes : K

éléments de fichiers (*chunks*) ou K différentes descriptions d'un même flux vidéo. La qualité du service perçue par un utilisateur dépend du nombre de données distinctes qu'il reçoit. La demande $d(u)$ peut donc être formulée comme étant égale à K et correspond à la meilleure qualité possible. Chaque donnée est diffusée dans le système dans une arborescence, ainsi, plus formellement, la distribution des données est organisée en K arborescences T_k , pour tout entier k inférieur à K . Pour un nœud u appartenant à l'arborescence T_k , le nombre d'enfants qu'il possède dans cette arborescence est noté $m_k(u)$. Le problème d'optimisation de la qualité générale de service s'écrit alors comme suit.

PROBLÈME.– K -DCDA

Étant donné un overlay $G(V, E^)$, une source s et une fonction de distribution de capacité $c(u)$ pour tout u dans V , il s'agit de déterminer K fonctions d'allocations w_k de E^* vers $\{0, 1\}$, pour tout k inférieur à K , telle que la somme des qualités de service est maximisée. Les arborescences T_k dont la racine est en s et formées par les arcs ayant un poids non-nul dans w_k doivent respecter que, pour chaque nœud u , la somme du nombre d'enfants de u dans toutes les arborescences est inférieure à la capacité de u , c'est-à-dire $\sum_{k \leq K} m_k(u) \leq c(u)$.*

A noter que, dans le problème de K -DCDA, la demande est considérée comme une fonction booléenne, ce qui représente une simplification significative du problème de maximisation d'utilité telle qu'il est décrit dans la littérature [CHI 07]. Ici, nous ne cherchons pas à optimiser un critère en recouvrant l'ensemble des nœuds d'un graphe, un des problèmes les plus étudiés dans la littérature scientifique. L'idée consiste plutôt à maximiser le nombre de nœuds qui sont couverts en respectant des contraintes.

Nous avons défini deux problèmes parmi beaucoup d'autres problèmes d'optimisation qu'il serait possible d'envisager sur un modèle aussi générique. Nous allons ensuite décrire les principaux résultats théoriques obtenus autour de ces problèmes.

1.3. Allocation optimale dans un cas générique

L'objectif de cette partie consiste à déterminer une allocation de ressources qui maximise la quantité de ressources allouées de pairs vers les pairs, avec le respect de la capacité et de la demande. Comme nous allons le montrer plus loin, une telle allocation des ressources peut être calculée en temps polynomial. Ce résultat est obtenu en transformant le problème SRA en un problème de flot maximal dans un graphe dont les liens sont pondérés. Une telle transformation n'est pas inhabituelle, mais le problème résolu ici n'a été démontré que récemment. Le calcul de flot maximal dans un graphe peut être résolu par des algorithmes polynomiaux classiques tels que celui de Ford-Fulkerson ou de Edmonds-Karp.

1.3.1. Transformation en un problème de flot

Nous concevons un réseau $N(G, c, d) = (V', E', w)$ à partir de l'overlay $G(V, E^*)$ et des fonctions de distribution des capacités $c(u)$ et des demandes $d(u)$. La Figure 1.1 illustre une telle transformation sur un exemple simple. L'ensemble V' des éléments du réseau contient un puits p , une source s et, pour chaque pair u dans V , nous créons deux sommets u^+ et u^- . Soient V^+ et V^- respectivement l'ensemble des nœuds u^+ et l'ensemble des nœuds u^- pour tout nœud u dans V . Plus formellement, nous obtenons que V' est l'union de V^+ avec V^- et les deux nœuds artificiels supplémentaires, la source s et le puits p .

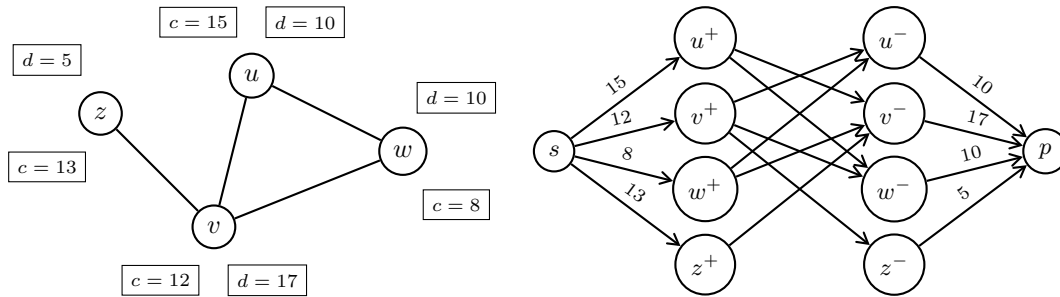


Figure 1.1. Transformation en un réseau d'un overlay contenant quatre paires.

L'ensemble des arcs E' comprend trois sous-ensembles distincts. Le première contient n arcs de la source s vers tous les nœuds dans V^+ . La capacité d'un arc (s, u^+) est égale à la capacité de ressource de u , c'est-à-dire $c(u)$. Le deuxième sous-ensemble comprend, à l'autre extrémité du réseau, n arcs de tous les nœuds dans V^- vers le puits p . Ici, la capacité de l'arc (u^-, p) est égale à la demande du nœud u . Enfin, dans le troisième sous-ensemble d'arcs, nous attribuons un arc de u^+ vers v^- s'il existe une arête (u, v) dans E^* . La capacité de cet arc est infinie. Ainsi l'ensemble d'arcs de ce réseau est formellement l'union de l'ensemble $\{(s, u^+), \forall u \in V\}$ avec l'ensemble $\{(u^-, p), \forall u \in V\}$ et l'ensemble $\{(u^+, v^-), \forall (u, v) \in E^*\}$. Une telle transformation est illustrée en Figure 1.1.

1.3.2. Allocation optimale de ressources

Une fois que ce réseau a été construit, il est facile de constater que le problème SRA consistant à déterminer l'allocation optimale peut trouver une solution en calculant le flot maximal dans ce réseau. Pour cela, soit f le flot maximal dans $N(G, c, d)$. Une fonction de poids w peut être définie comme suit : pour tout arc (u, v) dans E^* , le poids de cet arc peut être définie comme étant égale au flot circulant dans l'arc

(u^+, v^-) dans E' . Ainsi, déterminer l'allocation optimale de ressources du problème SRA devient équivalente à un calcul de flot maximal sur un graphe bipartite.

Dans un problème de flot maximal, l'objectif est de trouver la valeur maximale d'un flot entre une source unique et un puits unique alors que le réseau est caractérisé par des arêtes ayant des capacités maximales données. Deux célèbres algorithmes de calcul de la solution optimale dans de tels cas sont Ford-Fulkerson et Edmonds-Karp. Ces algorithmes admettent un temps de calcul en $O(|E| \cdot f)$ et $O(n \cdot |E|^2)$ respectivement, où n est le nombre de sommets du réseau et f est la valeur du flot.

Si les capacités des nœuds dépassent les demandes, alors la principale contrainte sur la valeur du flot maximal se situe au niveau des demandes puisque la capacité des liens à partir des nœuds dans V^+ vers les nœuds dans V^- est infinie. Cela signifie alors que la valeur du flot maximal est égale à la somme des demandes. Par conséquent, en définition de la conservation de flux, si la valeur de f est égale à la somme des demandes, on obtient que w atteint la demande maximale. Plus généralement, tout débit maximal f sur le réseau $N(G, c, d)$ permet de déterminer une fonction associée de poids w pour l'overlay G , de sorte que la demande pour tous les pairs est remplie si et seulement si la valeur de f est la somme des demandes. En d'autres termes, une réponse à la problème de décision peut être immédiatement déduite d'un calcul de flot maximal.

1.3.3. Discussion

En informatique théorique, la première question qui se pose concerne la complexité du problème, notamment s'il est possible de calculer une réponse au problème de décision associé au problème formel dans un temps raisonnable. De nombreux problèmes liés aux graphes ont révélé être NP-complet, c'est-à-dire qu'il n'existe pas, jusqu'à présent, d'algorithmes polynomiaux pour résoudre ces problèmes et, s'il en existait ne serait-ce qu'un, alors tous ces problèmes de cette famille seraient théoriquement solvables en temps polynomial. La plupart des scientifiques considèrent néanmoins que P est différent de NP et que la solution à un problème dit NP-complet nécessite un temps de calcul exponentiel en la taille du problème, autrement dit, il n'est pas raisonnable d'espérer obtenir une solution exacte à un problème de grande taille.

Le résultat présenté dans ce chapitre ne démontre pas que le problème SRA n'est pas NP-complet. En revanche, il démontre que, s'il est possible de déterminer une solution parfaite pour le problème, c'est-à-dire dans laquelle toutes les demandes sont satisfaites, alors il est possible de déterminer en un temps polynomial cette solution. Le résultat obtenu dans un cas où toutes les demandes ne peuvent pas être satisfaites est sans doute pas mauvais, mais il n'est pas forcément optimal. Par ailleurs, déterminer une solution bornée, c'est-à-dire dans laquelle chaque nœud admet un nombre

limité de voisins vers lesquels des ressources ont été alloués, s'est révélé être un problème NP-complet [LEB 08]. L'approche qui a été présentée dans ce chapitre ne peut donc pas s'appliquer si une contrainte supplémentaire consiste à limiter le nombre de voisins vers lesquels chaque pair affecte ses ressources. Or, une telle contrainte apparaît souvent dans les réseaux pair-à-pair. Le problème de déterminer une allocation de ressources qui maximiserait les ressources allouées tout en limitant le degré du graphe utilisé est donc toujours un problème ouvert.

Par ailleurs, et pour aborder des considérations plus pratiques, plusieurs algorithmes ont été présentés dans un cadre très générique pour apporter une solution au problème de déterminer un flot maximal de manière distribuée. Bien entendu, il est possible d'exécuter ces algorithmes pour que chaque pair puisse participer au calcul de l'allocation optimale. Ainsi, il est possible aujourd'hui de construire facilement un protocole pair-à-pair qui optimise l'allocation de ressources dans le cadre particulier du régime stationnaire introduit ici. En revanche, il est clair que ces algorithmes ayant été conçus pour obtenir une solution sur n'importe quel type de graphe, ils ne sont pas du tout optimisés pour les graphes très particuliers qui résultent de la transformation de l'overlay en réseau. Sans augurer de travaux futurs, il est sans doute possible de concevoir des algorithmes pair-à-pair qui utilisent de manière plus efficace les caractéristiques de ces réseaux.

1.4. Optimisation d'allocation de ressources dans un cas concret

Nous incluons maintenant les biens non rivaux dans le problème d'allocation des ressources, et nous calculons la capacité du système sous la contrainte de la disponibilité des données. Un pair ne peut pas allouer sa bande passante s'il ne dispose pas des données à transmettre. Les biens non-rivaux sont un ensemble comprenant K unités indépendantes de données (des *chunks*, des couches vidéo ou des descriptions). Nous considérons pour simplifier le problème que les données sont à peu près équivalentes en taille, ce qui est vrai pour le cas de *chunks* ou de descriptions de vidéos. La qualité de service associée à un pair est alors une fonction du nombre de données reçues.

Chaque élément de données k , $1 \leq k \leq K$ est servi aux nœuds sur une arborescence séparée $T_k = (V_k, E_k)$, un arbre dont la racine est s et où $V_k \subseteq V$ et $E_k \subseteq E^*$. Les enfants d'un pair $u \in V_k$ sont notés $N_k(u)$, tandis que le nombre d'enfants est noté $m_k(u)$. La construction des multiples arbres doit prendre en compte la contrainte sur la capacité d'un nœud u , soit $\sum_{k \in K} m_k(u) \leq c(u), \forall u \in V$.

Comme indiqué dans le problème K -DCDA, nous visons à maximiser la qualité globale du service dans l'overlay. Ici, nous définissons cette quantité comme étant la somme des qualités de service perçues par tous les clients. Notre modèle peut prendre en compte différentes définitions de cette qualité globale du service, par exemple assurer l'équité entre les clients ou maximiser le nombre de clients pour qui la qualité

expérimentée est supérieure à un certain seuil de qualité. Comme nous allons le montrer ci-dessous, ce problème d'optimisation est néanmoins NP-complet, même pour une valeur $K = 1$.

1.4.1. Preuve de NP-complétude

Une formulation formelle du problème de décision en rapport avec K -DCDA est :

INSTANCE.– Un graphe $G = (V, E^*)$ avec V l'ensemble des sommets et E^* l'ensemble des arêtes, une racine $s \in V$, un entier positif K , une fonction de capacité $c : V \rightarrow N$ et un entier positif Γ .

QUESTION.– Existe-t-il K arborescences $(T_k = (V_k, E_k))_{1 \leq k \leq K}$ avec pour racine s tel que :

- 1) pour tout k , nous avons $V_k \subseteq V$ et si (u, v) est une arête de T_k alors (u, v) est une arête de G ,
- 2) pour tout sommet $u \in V$, la somme du nombre de ses enfants dans toutes les arborescences est inférieure ou égale à sa capacité, soit $\sum_{k=1}^K m_k(u) \leq c(u)$,
- 3) le nombre total de sommets appartenant aux arborescences est supérieur ou égal à Γ , soit $\sum_{k=1}^K |V_k| \geq \Gamma$.

Le problème de décision associé au K -DCDA cherche l'existence d'une solution de K -DCDA où le nombre de nœuds desservis atteint au moins un paramètre fixe. Nous montrons que ce problème de décision est NP-complet. Pour des raisons de concision, nous fournissons seulement un aperçu de la preuve, la preuve complète apparaît dans [CHA 09].

THÉORÈME.– K -DCDA est NP-complet, même pour une valeur $K = 1$

Aperçu de la preuve : Le problème de *Bounded Degrée Spanning Tree* (BDST) est connu comme étant un problème NP-complet [GAR 79]. Étant donné un graphe $G = (V, E)$ et un entier positif $m \leq |V|$, le BDST consiste en déterminer s'il existe un arbre couvrant pour G dans lequel aucun sommet n'a plus de m enfants. Considérons maintenant une instance du problème K -DCDA dans lequel $K = 1$, dans lequel $c(u) = m, \forall u \in V$, et dans lequel le nombre de nœuds desservis doit être n . Par ailleurs, la source s peut être n'importe quel nœud dans V . Il est trivial d'observer qu'une solution de cette instance du K -DCDA est aussi une solution du BDST, et vice-versa.

1.4.2. Modèle en nombres entiers

Nous formulons le problème décrit ci-dessus avec un modèle de programmation en nombres entiers. Nous considérons d'abord un cas plus simple où $K = 1$, puis

nous étendrons la formulation au cas général. Notez que, dans le 1-DCDA, l'objectif est de maximiser le nombre de nœuds servis, la qualité du service étant une fonction booléenne. Le 1-DCDA s'établit comme suit. Pour chaque nœud u et pour chaque nœud $v \in N(u)$, la variable de décision x_{uv} est égal à 1 si u envoie la donnée à v , et 0 autrement. La fonction à optimiser peut s'écrire formellement comme suit :

$$\max \sum_{u \in V} \sum_{v \in N(u)} x_{uv}, \quad \text{tel que}$$

$$\sum_{v \in N(u)} x_{uv} \leq 1, \forall u \in V \quad (1.1)$$

$$\sum_{u, v \in S} x_{uv} \leq |S| - 1, \forall S \subseteq V \quad (1.2)$$

$$\sum_{v \in N(u)} x_{uv} \leq c(u), \forall u \in V \quad (1.3)$$

$$x_{uv} \leq \sum_{w \in N(u)} x_{wu}, \forall u \in V \setminus \{s\}, \forall v \in N(u) \quad (1.4)$$

$$x_{uv} \in \{0, 1\}, \forall u \in V, \forall v \in N(u) \quad (1.5)$$

La contrainte (1.1) requiert qu'un nœud puisse recevoir une description d'au plus un nœud et la contrainte (1.2) exige que les arcs ne forment pas un cycle. Ces deux contraintes sont classiquement utilisées pour assurer la construction d'une arborescence. La contrainte de capacité du nœud apparaît dans la contrainte (1.3) où le nombre d'enfants d'un nœud est limité par sa capacité. Enfin, la contrainte (1.4) assure qu'un nœud peut envoyer la donnée, si et seulement si il l'a reçue d'un autre nœud. Dans cette formulation, les variables introduites en (1.5) sont liées aux arêtes, nous avons donc $2 \cdot m$ variables.

Nous étendons maintenant ce modèle au cas générique où K est un nombre entier positif non-nul. La formulation de l'objectif dépend alors de la qualité de service perçue par les nœuds. Pour simplifier, nous notons $q(k)$ la fonction continue croissante qui associe une qualité perçue à un nombre de données reçues. La fonction à optimiser devient alors :

$$\sum_{u \in V} q \left(\sum_{v \in N(u)} \sum_{k=1}^K x_{uv}^k \right), \quad \text{tel que}$$

Les contraintes ne sont pas fondamentalement différentes de la formulation précédente.

$$\begin{aligned}
\sum_{v \in N(u)} x_{uv}^k &\leq 1, \forall u \in V, 1 \leq k \leq K \\
\sum_{u, v \in S} x_{uv}^k &\leq |S| - 1, \forall S \subseteq V, 1 \leq k \leq K \\
\sum_{k=1}^K \sum_{v \in N(u)} x_{uv}^k &\leq c(u), \forall u \in V \\
x_{uv}^k &\leq \sum_{w \in N(u)} x_{uw}^k, \forall u \in V \setminus \{s\}, \forall v \in N(u), 1 \leq k \leq K \\
x_{uv}^k &\in \{0, 1\}, \forall u \in V, \forall v \in N(u), 1 \leq k \leq K
\end{aligned}$$

La variable booléenne x_{uv}^k est maintenant égale à 1 si u envoie la donnée k à v , et 0 autrement. Notez que les contraintes (1.1), (1.2) et (1.4) sont simplement dupliqués K fois. En revanche, la contrainte (1.3) nécessite d'examiner tous les arbres pour effectuer le calcul des capacités. Ici, le modèle se base sur $2 \cdot k \cdot m$ variables et $K \cdot n + n + 2 \cdot K \cdot m$ contraintes.

Dans ce modèle, veuillez noter qu'il n'existe pas de préférence entre les K données. Cette précision est essentielle car elle apporte une symétrie dans la conception des solutions : une solution optimale peut être obtenue en remaniant les valeurs des données. Cette symétrie devrait logiquement être supprimée de la formulation, afin d'accélérer le calcul de la solution.

1.4.3. Discussion

Le cas du régime stationnaire s'était révélé ne pas être NP-complet pour le cas où toutes les demandes des pairs pouvaient être satisfaites. Dans le cas où l'allocation des ressources dépend de l'acquisition d'une donnée, nous avons montré que l'optimisation de l'allocation de ressources est NP-complet. Autrement dit, il est fort probable que la solution optimale ne puisse pas être calculée dans un délai raisonnable. Nous parlons ici de calcul centralisé, c'est-à-dire dans le cas où un opérateur de réseau pair-à-pair pourrait avoir une vision globale du système, ce qui n'est pas toujours réaliste. Ce résultat est donc une très mauvaise nouvelle.

Dans le même temps, nous avons présenté un modèle de programmation en nombres entiers qui est prometteur. En effet, un bon modèle en nombres entiers permet tout d'abord de calculer des solutions exactes dans des temps un peu plus raisonnables qu'une approche systématique, ensuite il est à la base de très nombreuses techniques

permettant de concevoir des algorithmes d'approximation, c'est-à-dire des algorithmes capables de calculer en temps polynomial une solution qui est, au pire, un facteur de la solution exacte optimale.

D'un point de vue pratique, ce modèle permet également de comparer sur des petites instances des solutions obtenues par un algorithme distribué basé sur un protocole pair-à-pair avec des solutions optimales. C'est par exemple le cas dans [CHA 09]. L'autre utilisation potentielle est l'étude sur des petits cas de la performance d'overlays. Certains overlays se prêtent plus ou moins bien à une allocation performante des ressources. Ainsi, un concepteur d'overlays pair-à-pair peut hésiter au moment de spécifier le nombre de voisins des pairs dans un overlay. Avec un comparatif entre plusieurs propositions, il peut être capable de déterminer le potentiel des overlays pour toutes ces propositions. En revanche, il n'est pas forcément capable aujourd'hui d'exploiter totalement ce potentiel.

1.5. Littérature dans ce domaine

Le problème de la capacité des réseaux pair-à-pair est un sujet d'étude relativement récent et encore largement inexploré. Dans ce sens, le présent chapitre décrit principalement les premiers résultats théoriques ainsi que le modèle qui possède le potentiel de recherche le plus prometteur à ce jour.

La plupart des travaux récents ont porté sur la diffusion de flux vidéo en direct. Par exemple, le papier [JIN 07] propose un modèle d'overlay dans lequel les contraintes de capacité apparaissent dans les liens. Les auteurs montrent que déterminer la bande-passante allouable maximale dans ce modèle est NP-complet. Ce résultat n'est pas très surprenant puisqu'il aurait pu se situer dans le domaine scientifique, nettement plus étudié, de conception réseau [JOH 78] ou de partages de ressources dans un réseau dans lequel les processeurs sont modélisés comme des files d'attente [ROB 04]. Plusieurs travaux récents ont montré que ces modèles, encore valides dans un contexte d'Internet, ne sont pas très pertinents dans le domaine des réseaux pair-à-pair dans lesquels la contrainte de capacité d'accès des pairs au réseau est importante tandis que la principale limitation provient de la capacité d'offrir des ressources des pairs vers les autres pairs.

Ce modèle de réseau similaire a été utilisé dans [LIU 08], dans lequel les auteurs présentent plusieurs variantes du problème du calcul de l'allocation maximale de bande passante à tous les pairs dans le réseau. Une approximation de la solution optimale, basée sur l'utilisation de technique générique autour de modèle de programmation linéaire est présentée. Les auteurs démontrent que cette approximation peut s'appliquer directement à tous les cas étudiés, sauf lorsque les nœuds ont des degrés sortants limités. De manière surprenante, les auteurs ne cherchent pas à montrer préalablement que les problèmes sont NP-complet. De fait, il a été montré par la suite, tel

que décrit dans ce chapitre, que, dans de nombreux cas des solutions exactes pouvaient être obtenues en un temps polynomial. Néanmoins, ce rapport technique est d'une grande richesse et il apporte un éclairage d'une rare pertinence sur les problèmes d'optimisation d'allocation de ressources dans les réseaux pair-à-pair lorsque les capacités des pairs sont insuffisantes. Une approche similaire basée sur le dual d'une modélisation par programmation entière dans le cas où le réseau admet principalement des limitations sur les liens est également montré dans [CUI 04].

Par ailleurs, plusieurs études ont cherché à analyser la performance des réseaux pair-à-pair pour le transfert de fichier d'un expéditeur à de nombreuses destinations. Un travail particulièrement pionnier dans ce domaine est [YAN 06]. La plupart des études suivantes ont porté sur l'analyse des performances de stratégies d'ordonnancement, c'est-à-dire quel est le temps nécessaire pour transmettre un fichier à n clients dans le réseau. Un des résultats les plus marquants est publié dans [QIU 04]. Ce papier décrit un modèle fluide relativement simple pour analyser la performance du système pair-à-pair *bit-torrent*. Cependant, ces travaux considèrent que les nœuds ont une vue complète du système, ce qui semble très optimiste dans un cas réaliste. En outre, ces travaux assument généralement que les pairs connaissent les stratégies d'ordonnancement de leurs voisins, ce qui ne semble pas facile à obtenir concrètement. Ces travaux sont néanmoins une grande source d'inspiration pour tous les chercheurs scientifiques désireux d'acquérir une culture en mathématique pour les réseaux pair-à-pair.

De manière plus générale, le problème consistant en déterminer une allocation de ressources maximale est relié à l'immense littérature autour de la construction d'arbres optimaux. Parmi les résultats les plus proches des problèmes décrits dans ce chapitre, veuillez noter que déterminer un arbre couvrant de degré borné (BDST) en l'absence de nœud ayant plus de m enfants est un problème NP-complet pour n'importe quel degré m supérieur à 2. Le BDST s'avère être un cas particulier du problème 1-DCDA pour lequel la capacité des nœuds serait fixée, pour tout nœud u , à une valeur $c(u)$ égale à m . De nombreuses études ont dérivé de ce résultat fondamental, notamment des études sur les arbres couvrants ayant un coût minimal sur un graphe pondéré [GOE 06]. Des variations dans l'esprit de ce qui a été montré dans ce chapitre incluent notamment l'étude du problème dans le cas où les degrés bornés ne sont pas uniformes [KÖN 05] où lorsque la profondeur de l'arbre couvrant doit être minimal [HEL 07].

La formulation présentée dans ce chapitre est somme toute assez proche de ces travaux, mais il diffère en deux façons qu'il nous semble intéressant de pointer. Tout d'abord, l'overlay considéré est un graphe non-pondéré sur les liens, ce qui représente finalement une différence très sensible sur le critère d'optimisation. Ensuite, l'immense majorité de ces travaux visent à couvrir l'ensemble des nœuds tout en minimisant par ailleurs un ou plusieurs critères donnés alors que l'optimisation dans les systèmes pair-à-pair sous-provisionnés vise plutôt à satisfaire le maximum de participants.

Parmi les travaux théoriques qui s'intéressent de plus près à cette dernière remarque, nous notons le papier [BLU 05] qui vise à construire un arbre couvrant au moins k nœuds parmi les n nœuds dans un graphe pondéré. Il n'est pas donné d'algorithmes permettant de déterminer le nombre k maximal.

Quand un réseau est modélisé sous la forme d'un graphe avec des arcs associés à des poids et les nœuds à des gains, il est possible de formuler un problème d'optimisation dans lequel le bénéfice des nœuds connectés soustrait du poids des arcs utilisés cherche à être maximisé. La famille de problèmes s'appelle le *Price-Collecting Steiner Tree Problem* (PCSTP) [VOS 06]. Il est facile de montrer que le problème 1-DCDA décrit dans ce chapitre peut être ramené à PCSTP puisque 1-DCDA cherche à maximiser le nombre de nœuds attachés, ce qui est équivalent à maximiser le bénéfice lorsque les poids sont nuls. Toutefois, 1-DCDA demeure différent puisque les nœuds ont une contrainte sur le degré.

Enfin, de nombreux travaux ont porté sur la conception d'algorithmes visant à construire une application pair-à-pair pour réaliser un protocole de communication de groupe (multicast) sur une couche haute (voir par exemple [HOS 07] pour un état de l'art). L'objectif est encore de couvrir tous les participants dans l'overlay, mais les objectifs d'optimisation sont ici basés sur des considérations beaucoup plus réalistes (par exemple réduire la surcharge liée aux messages de contrôle ou à gérer les défaillances de manière rapide). Plusieurs algorithmes ont été proposés et largement analysés par simulations (voir [FAH 07] pour une étude approfondie). Les travaux les plus connus sont ZigZag et Nice [TRA 04] dans lesquels les pairs tendent à s'organiser en grappes pour réduire la charge des messages de contrôle dans le maintien de l'arbre multicast. Dans la même veine, TAG [KWO 05] prend en compte la topologie du réseau sous-jacent pour construire plusieurs arbres de diffusion

1.6. Conclusion

La communauté scientifique commence à étudier plus particulièrement le problème d'allocation de ressources dans les overlays. En effet, ces topologies étant de plus en plus contraintes par des paramètres externes, il est de moins en moins possible de modifier la structure de l'overlay pour optimiser les échanges entre pairs. Dans ce cadre, un modèle intéressant consiste en séparer les ressources qui sont effectivement contraintes, les ressources rivales telles que la bande-passante montante, des biens non-rivaux, par exemple les données. Dans ce modèle, il est possible de distinguer deux cas : le cas où les biens non-rivaux peuvent être négligés, c'est-à-dire le cas du régime stationnaire dans *bit-torrent*, et le cas où les biens non-rivaux conditionnent l'efficacité de l'allocation de ressources, typiquement le cas de la diffusion d'un flux vidéo.

Dans ce chapitre, nous sommes revenus sur les premiers résultats théoriques publiés jusqu'ici dans ce domaine. Dans le régime stationnaire, nous avons montré qu'il

est théoriquement possible de déterminer une allocation optimale de ressources relativement facilement, et ce de manière distribuée. En revanche, lorsque les données conditionnent l'allocation des ressources, le problème d'optimisation devient insoluble pour des systèmes de grande taille, même pour un opérateur ayant une vision globale du système.

Ces deux résultats théoriques ouvrent des perspectives intéressantes pour la conception de protocoles pair-à-pair dans lesquels l'allocation de ressources cherche à être plus intelligente que ce qui a été fait jusqu'à présent. Typiquement dans *bit-torrent*, un pair divise sa bande-passante montante par le nombre de pairs qu'il serve. Suite aux résultats présentés ici, il est possible qu'un pair puisse allouer en parts inégales sa bande-passante afin d'améliorer globalement la transmission des données dans le système.

1.7. Bibliographie

- [BLU 05] BLUM C., BLES A. M. J., « New metaheuristic approaches for the edge-weighted -cardinality tree problem », *Computers & OR*, vol. 32, p. 1355-1377, 2005.
- [CHA 09] CHAKARESKI J., FROSSARD P., KERIVIN H., LEBLET J., SIMON G., A note on the data-driven capacity of p2p networks. Rapport n°EPFL-LTS-2009-008, École Polytechnique Fédérale de Lausanne, July 2009.
- [CHI 07] CHIANG M., LOW S., CALDERBANK A., DOYLE J., « Layering as Optimization Decomposition : A Mathematical Theory of Network Architectures », *IEEE Proceedings*, vol. 95, n° 1, page255, 2007.
- [CUI 04] CUI Y., LI B., NAHRSTEDT K., « On achieving optimized capacity utilization in application overlay networks with multiple competing sessions », *Proc. of ACM Symp. on Parallelism in Algo. and Archi. (SPAA)*, p. 160–169, 2004.
- [FAH 07] FAHMY S., KWON M., « Characterizing overlay multicast networks and their costs », *IEEE/ACM Transactions on Networking*, vol. 15, n°2, p. 373-386, 2007.
- [GAR 79] GAREY M. R., JOHNSON D. S., *Computers and intractability*, WH Freeman San Francisco, 1979.
- [GOE 06] GOEMANS M. X., « Minimum bounded degree spanning trees », *Proc. of IEEE Symp. on Foundations of Comp. Sci. (FOCS)*, p. 273–282, 2006.
- [HEL 07] HELMICK M. T., ANNEXSTEIN F. S., « Depth-Latency Tradeoffs in Multicast Tree Algorithms », *Proc. of IEEE Int. Conf. on Advanced Information Networks and Applications (AINA)*, p. 555-564, 2007.
- [HOS 07] HOSSEINI M., AHMED D. T., SHIRMOHAMMADI S., GEORGANAS N. D., « A Survey of Application-Layer Multicast Protocols », *IEEE Communications Surveys & Tutorials*, vol. 9, n°3, p. 58–74, 2007.

- [JIN 07] JIN X., YIU W.-P., CHAN S.-H., WANG Y., « On Maximizing Tree Bandwidth for Topology-Aware Peer-to-Peer Streaming », *IEEE Transactions on Multimedia*, vol. 9, n°8, p. 1580-1592, Dec. 2007.
- [JOH 78] JOHNSON D., LENSTRA J., KAN A., « The complexity of the network design problem », *Networks*, vol. 8, n°4, p. 279-285, 1978.
- [KÖN 05] KÖNEMANN J., RAVI R., « Primal-Dual Meets Local Search : Approximating MSTs With Nonuniform Degree Bounds », *SIAM J. Comput.*, vol. 34, n°3, p. 763-773, 2005.
- [KWO 05] KWON M., FAHMY S., « Path-aware overlay multicast », *Computer Networks*, vol. 47, n°1, p. 23-45, 2005.
- [LEB 08] LEBLET J., PIANESE F., SIMON G., « Optimizing Resource Sharing in Node-Capacitated Overlay Networks », *Proc. of Autonomous and Spontaneous Networks Symposium (ASN)*, 2008.
- [LIU 08] LIU S., SENGUPTA S., CHEN M., LI J., CHIANG M., CHOU P. A., Streaming Capacity in Peer-to-Peer Networks with Topology Constraints, Rapport, Princeton University, Sep. 2008.
- [MAS 07] MASSOULIÉ L., TWIGG A., GKANTSIDIS C., RODRIGUEZ P., « Randomized Decentralized Broadcasting Algorithms », *Proceedings of 26th IEEE International Conference on Computer Communications (INFOCOM)*, p. 1073-1081, 2007.
- [QIU 04] QIU D., SRIKANT R., « Modeling and performance analysis of BitTorrent-like peer-to-peer networks », *ACM SIGCOMM*, 2004.
- [ROB 04] ROBERTS J. W., « A survey on statistical bandwidth sharing », *Comput. Netw.*, vol. 45, n°3, p. 319-332, 2004.
- [TRA 04] TRAN D. A., HUA K. A., DO T. T., « A peer-to-peer architecture for media streaming », *IEEE Journal on Selected Areas in Communications*, vol. 22, n°1, p. 121-133, 2004.
- [VAS 09] VASSILAKIS D., VASSALOS V., « An analysis of peer-to-peer networks with altruistic peers », *Peer-to-Peer Networking and Applications*, vol. 2, n°2, p. 109-127, 2009.
- [VOS 06] VOSS S., « Steiner Tree Problems in Telecommunications », RESENDE M. G. C., PARDALOS P. M., Eds., *Handbook of Optimization in Telecommunications*, p. 459-492, Springer US, New York, 2006.
- [YAN 06] YANG X., DE VECIANA G., « Performance of peer-to-peer networks : Service capacity and role of resource sharing policies », *Performance Evaluation*, vol. 63, n°3, p. 175-194, 2006.