Institut Eurécom
Department of Corporate Communications
2229, route des Crètes
B.P. 193
06904 Sophia-Antipolis
FRANCE

Research Report RR-08-210

# General Resource Sharing Problems in Overlay Networks

Jimmy Leblet, Pietro Michiardi, Gwendal Simon[1]

Tel : (+33) 4 93 00 81 45
Fax : (+33) 4 93 00 26 27
Contact at Eurecom : {Pietro.Michiardi}@eurecom.fr

[1] Institut Eurécom's research is partially supported by its industrial members: Bouygues Téécom, Orange, Hitachi Europe, SFR, Sharp, ST Microelectronics, Swisscom, Thales, BMW, CISCO.

[1] Computer Sc. Dept - Institut Telecom / Telecom Bretagne - France - {jimmy.leblet,gwendal.simon}@telecom-bretagne.eu

**Abstract**

In this paper we propose a theoretical framework to analyze and assess the properties of overlay networks with the goal of studying problems related to rival resource sharing. We study optimality goals for the construction of such overlay networks and focus on fairness issues by defining perfectly reciprocal overlays, in which every peer receives exactly the same amount of resources it offers. We focus on answering the following key question : is it possible to determine whether a perfectly reciprocal overlay exists for a given physical network configuration? First, we prove that deciding whether there exists a perfectly reciprocal overlay given a fixed underlay can be verified in polynomial time. We also propose an algorithm to compute an optimal overlay network, if one exists. Second, we prove that the same decision problem becomes NP-complete if one looks for bounded degree overlays. This last result is of paramount importance because overlay designers frequently constrain peer outdegree. Finally, we present a practical application of our framework: throughout numerical simulations we assess the optimality of overlay networks constructed using distributed algorithms akin to stable matching theory.

# 1 Introduction

The remarkable scalability that characterizes peer-to-peer applications, for example content sharing applications such as BitTorrent [16], could not be achieved without users cooperation. A key requirement is especially users' contribution with private, often limited, resources such as CPU, storage and bandwidth. Hence, selfish behavior and free-riding have often emerged as a real problem from the earliest examples of such cooperative systems [10]. Because the total amount of tangible and rival resources is limited, an increase in the number of users without a corresponding raise in the quantity of shared resources would make each user pay the price of other users' selfishness.

Some techniques have proven to be efficient in ensuring a fair amount of resource reciprocation among a large set of peers, for example tit-for-tat-like enforcement mechanisms [13]. However, the question of whether such mechanisms achieve *optimal* performance (in the sense of perfect reciprocity) remain unanswered. Furthermore, it is not well understood if, given a particular communication graph linking users, each characterized by peculiar resource constraints, it is possible to designate logical neighborhood relationships that achieve perfect reciprocity. Indeed, despite a large literature related with peer-to-peer networks, little has been done for the theoretical assessment of the "fairness" properties in overlay networks. Few formal definitions can be associated with optimality goals or impossibility results.

We propose in this paper a theoretical framework so that a large set of problems related with rival resource sharing in overlay networks can be studied. The proposed model distinguishes (i) the network supporting the communication between peers, that is called *underlay*, (ii) the *rival resources* each peer is giving and receiving to and from the peer-to-peer system and (iii) the set of pair of peers actually exchanging rival resources, which is commonly known as the *overlay*. Although this model is based on strong assumptions, we argue it matches reasonably well with current networking reality. Furthermore, its relative simplicity allows us to identify four distinct problems whether the overlay or the resource distribution is known or not. A suggestion of various optimization objectives completes this generic and theoretic structure which constitutes the first contribution of this paper. All details are given in Section 3.

The rest of the paper deals with *perfectly reciprocal overlays* where each peer receives exactly the same amount of rival resources that it offers. Two arguments motivate this focus: perfectly reciprocal overlays are also optimal for the total amount of resources exchanged within the system and ensuring a perfect reciprocity to peers may be a key incentive for them to contribute. We aim to determine whether a perfectly reciprocal overlay can exist on top of a given network configuration. We prove two strong results. First, deciding whether there exists a perfectly reciprocal overlay given a fixed underlay can be verified in polynomial time. We propose an algorithm based on an elegant augmentation of the underlay graph. This algorithm also gives one overlay if a solution exists. Second, the same decision problem becomes NP-complete if one looks for bounded-outdegree overlays. This last result is of primary importance because most distributed algorithms for peer-to-peer systems make a peer connect with a limited number of neighbors. This noteworthy contribution is described in Section 4.

Finally, Section 5 shows a case study using this theoretical framework. Indeed, if this paper does not give any distributed algorithm for building perfectly reciprocal overlays, it provides a complete set of algorithms to evaluate them. We focus on overlays obtained using distributed techniques akin to $b$-matching [12] and show, using our framework, that in some cases $b$-matching-based overlay construction algorithms fails in building perfectly reciprocal overlays, but that these failure cases are very infrequent when the underlay is a random graph. The last Section concludes this paper.

## 2 Related Works

A family of works closely related to ours address the un-coordinated creation of graph topologies by selfish agents wishing to build communication networks. The seminal work by Fabrikant *et. al.* [6] study the design of the Internet graph in which selfish agents mimic the behavior of Autonomous Systems in building the network. In their model, called the *network creation game*, the authors analyze the conflicting goals of constructing a network which minimizes the construction costs while at the same time making sure its design is optimal (from a selfish point of view) with respect to some quality of service metrics. When the quality of service metric is expressed in terms of distance (for example number of hops) between any two nodes, the authors of [6] show the conditions under which a *stable* network can emerge and what is its performance when compared to a network created by a centralized entity whose goal is to maximize the aggregated quality of the network (they study the so-called *price of anarchy*).

Another work related to the creation of communication networks is due to Anshelevich *et. al* [3, 2]: in their work, called the *global connection game*, agents are allowed to build edges of a graph with a global vision of the resulting topology. The goal of the players of this game is to maximize the quality of the final network without any constraint on uniform connectivity: each player has a subset of nodes to connect to and a budget that it wishes to utilize the least. Each player evenly shares the cost of an edge and [2] show conditions (related to the Shapley [14] value) for the existence of such networks.

In [17] Wang and Li focus on the specific problem of bandwidth allocation in peer-to-peer overlay networks and propose a market model to match providers and consumers of the bandwidth resource. In their model, providers compete to attract consumers by adapting the resource price, while consumers seek at minimizing their costs. They also provide a decentralized algorithm based on reinforcement learning that achieves an equilibrium wherein nodes use their optimal strategies and the resource is near-optimally shared.

Authors in [15] target the creation of an overlay network for an $n$-way broadcast application in which each one of n overlay nodes wants to push large data files to all other $n-1$ destinations while downloading their respective data files. The authors design the Max-Min and Max-Sum algorithms used by individual nodes to select their neighbors: using these algorithms overlay nodes maximize the minimum and the aggregate network flows to all possible destinations. The work in [15] belong to the body of works that deal with *selfish neighbor selection* [11], which extend the basic concepts developed in [6].

Our theoretical framework can be used to analyze the "quality" of solutions achieved by distributed algorithms that address some specific resource sharing problems. Several works, including [8], [9] and [12], have modeled the creation of an overlay graph used for resource sharing as a stable matching problem and designed distributed algorithms to find such overlays. In this paper we present a case study which study, using a custom simulatior, $b$-matching based algorithms and assess their optimality under different settings.

# 3 Theoretical Framework

## 3.1 Model and Notations

We denote by $V$ the set of peers that belong to the peer-to-peer system and by $E$ the set of *possible* connections between two peers. A possible connection between two peers $x$ and $y$, noted $\{x, y\}$, indicates the existence of a communication link between $x$ and $y$ over which a generic resource can be shared. The couple $G = (V, E)$ defines the **underlay graph** of the system. In general, peer-to-peer systems designed for the Internet assume $G$ to be a clique, *i.e.* every peer may communicate and access resource shared by every other peer in the network. We note however that the physical infrastructure over which the peer-to-peer systems operates is not necessarily complete: for instance this is not the case for local area networks, where a large fraction of peers are placed behind firewalls or NATs or for applications where latency issues would prevent distant peers to connect. These topologies are covered by a general model as an underlay graph. For a peer $x$, the set of peers with which $x$ shares a links in $E$ are its underlay neighbors, denoted by $N(x)$ where $N(x) = \{y \in V : \{x, y\} \in E\}$.

At a first glance, undirected graphs depicting underlay neighborhood relations between peers seem appropriate. However, a peer $x$ could offer resources to a peer $y$ while conversely $y$ would choose to not reciprocate. This means that an edge between two peers $x$ and $y$ could be used through $\overrightarrow{xy}$ when $x$ provides resources to $y$, but not through $\overrightarrow{yx}$ when $y$ does not. We believe that undirected edges do not capture all resource sharing strategies between two peers in peer-to-peer systems. Therefore we formally define the underlay graph as $\overleftrightarrow{G} = (V, \overleftrightarrow{E})$ where $\overleftrightarrow{E} = \{\overrightarrow{xy} : \{x, y\} \in E\}$. This graph is just another way to express $G$ where each edge is replaced by the two possible oriented edges between vertices of the considered edge.

In most peer-to-peer systems relying on a very dense underlay graph, only a very small part of the edges in $\overleftrightarrow{E}$ is used. Furthermore, as current network protocols do not manage correctly too many simultaneous connections, it is common to bound the maximal number of edges from and to a peer in order to let networking resources be used efficiently [1]. We denote by $\overleftrightarrow{E}_U \subseteq \overleftrightarrow{E}$ the set of oriented edges that are actually used by the system. In this work we do not address *dynamic* peer-to-peer systems, hence $\overleftrightarrow{E}_U$ can be seen as a snapshot of a peer-to-peer system. We denote by $N_U^+(x)$ the set of peers which uses resources provided by $x$, thus $N_U^+(x) = \{y \in V : \overrightarrow{xy} \in \overleftrightarrow{E}_U\}$. In the same way, we denote by $N_U^-(x)$ the set of peers which provides resources to $x$, thus $N_U^-(x) = \{y \in V : \overrightarrow{yx} \in \overleftrightarrow{E}_U\}$. Peers $V$ and edges $\overleftrightarrow{E}_U$ form the **overlay graph**. Designing an overlay graph matching some detailed properties from a complete underlay has been the concern of a majority of studies dealing with peer-to-peer systems in recent years.

We introduce now the concept of **peer resource** by referring to the concept of good rivalry [4]. Most peer-to-peer applications care of the exchange of *non-rival* goods, for example electronic files in file-sharing systems. Indeed, sharing a file with another peer does actually not prevent its owner to use it. That is, these overlays aim to organize the global set of all non-rival resources provided by peers, so to handle abundance rather than to manage scarcity. However, peer-to-peer systems have also to deal with *rival* goods, *i.e.* scarce resources that are tangible such as bandwidth (*e.g.* in p2p video streaming applications), processing power (*e.g.* in grid computing systems) or storage capacity (*e.g.* in p2p storage and backup systems). Sharing scarce rival resources is the main focus of this paper. The amount of rival resources a peer $x$ is able to offer to other peers in the system is termed $\mathcal{R}(x)$. We further assume that rival resources can be subdivided into several sub-resources that are obviously rival too. This implies that we can define a function $\sum$ which that operates on two sub-resources and results in a resource unit.

For two peers $x$ and $y$, we denote by $w(\overrightarrow{xy})$ the amount of resources provided by $x$ to $y$. The overlay graph we study is then a weighted graph where each edge $\overrightarrow{xy}$ in $\overleftrightarrow{E}_U$ is associated with a non-null weight $w(\overrightarrow{xy})$. Previous definitions make a peer $y$ belonging to $N_U^+(x)$ if and only if $w(\overrightarrow{xy}) \neq 0$. A peer can not share more resource than it actually owns, formally $\mathcal{R}(x)$ is always greater than $\sum_{y \in N_U^+(x)} w(\overrightarrow{xy})$. In the same way, the amount of resources obtained by a peer $x$ is $\sum_{y \in N_U^-(x)} w(\overrightarrow{yx})$. Finally, the total amount of rival resources within the entire system is defined as $R = \sum_{x \in V} \mathcal{R}(x)$. Using these notation, we are able to introduce a **configuration** $C$ that consists in an underlay graph $\overleftrightarrow{G}$, an overlay graph $(V, \overleftrightarrow{E}_U, w)$ and a resource set $\mathcal{R} = \{\mathcal{R}(x) : \forall x \in V\}$. A configuration is said *feasible* when any peer $x$ verifies $\mathcal{R}(x) \geq \sum_{y \in N_U^+(x)} w(\overrightarrow{xy})$.

Note that the model introduced in this Section implies two strong albeit realistic assumptions. First, the weights assigned to overlay links depend on the resources owned by peers, hence metrics associated to the underlay network (*e.g.* available bandwidth, latency of a link, ...) are not taken in consideration when assigning a weight to an overlay link. We argue this assumption to hold in reality: it is typical to assume network bottlenecks to be located within the edge of the network, whereas the network core is often idealized. Second, we allow peers to consume an unlimited amount of resources. Although this assumption may be regarded as unrealistic, casual network connections of end-users are frequently asymmetric, *e.g.* xDSL users. Ignoring in a first instance any bottlenecks in resource consumption allows great simplification to our model.

### 3.2 Resource Sharing: Problem Formulation

The model allows to formulate distinct challenges expressed through a configuration $C$: underlay graph, overlay weighted graph and resource set. In this work we assume an underlay graph to be fixed while we let a centralized entity, termed the *overlay designer*, the task of designing an overlay graph capable of achieving optimal performance with respect to a family of metrics we detail further. The designer can be faced to problems of increasing difficulty, depending on the degrees of freedom available for the system design. We enumerate a set of possible problems, together with some realistic applications.

**Known Overlay Graph, Known Resource Distribution:** in this case the overlay graph is provided to the system designer and resources available to all participating peers follow a known distribution. However, given a link $\overrightarrow{xy}$ in $\overleftrightarrow{E}_U$, the link weight $w(\overrightarrow{xy})$ is assumed to be unknown. This problem refers to the task of evaluating the optimality of a given design, rather than the design itself. Here, the difficulty is to determine, whenever the edges belonging to a given overlay graph are used, an assignment of edge weights so that a global objective can be achieved. The set of solutions to this problem consists of weighted overlay graphs.

Several realistic scenarios may fall into this case. With the development of companies specialized in peer-to-peer systems, several overlays may actually compete for a market. The analysis of proposals or the evaluation of competitors calls for the definition of reliable and objective benchmarks aiming at evaluating the efficiency of an overlay structure. In this case, we study if a designer can determine whether (i) there exists an optimal solution to the problem and (ii) if a given overlay can achieve this optimal solution.

**Known Overlay Graph, Unknown Resource distribution:** as for the previous problem, the overlay graph $\overleftrightarrow{E}_U$ is known and fixed, but neither edge weights nor resource allocation are known. The problem here is to determine if a resource distribution which needs to be associated with edge weights exists so as to produce a feasible configuration achieving the global objective. Note that the solution space to this problem is infinite. Indeed, any new feasible configuration can be derived by multiplying both the amount of resources and the weights of edges of a feasible solution by any arbitrary fixed value.

Many practical situations make a designer face to this problem, that could appear unrealistic at a first glance. Consider for example an existing distributed system where one would like to upgrade some peers by increasing (or decreasing) the amount of allocated resources originally offered by these peers without modifying the logical structure that connects peers. Many structured systems let some exogenous parameters being primal in the choice of neighbors, the most popular example being Distributed Hash Tables.

**Unknown Overlay Graph, Known Resource Distribution:** here we assume a fixed and known set of resources but the overlay graph is to be built. A solution to this problem is a weighted overlay graph where a subset of links that belong to the underlay network are used to exchange a well defined amount of resources from one peer to the other. In other words, the question is to understand whether each peer can determine a resource assignment among all of its underlay neighbors that achieves optimal performance.

This problem has attracted a lot of attention: several works have been dedicated to the design of overlay construction mechanisms in which peers are assumed to freely join the system with a fixed resource budget which is not under the control of the overlay designer. However, most of previous work focused on the difficult task of handling peer churn and system dynamics. Instead, the optimality of the overlay structures achieved through these mechanisms has received little attention to the best of our knowledge.

**Unknown Overlay Graph, Unknown Resource Distribution:** in this problem we assume the only design constraint to be the underlay network graph, while the overlay graph, the resource distribution are assumed to be unknown. The question is whether it is possible to design an optimal overlay graph, together with an optimal assignment of resources which achieves perfect reciprocation.

This situation is very open, especially when the underlay is seen as a clique. Yet the underlay may be sometimes very constraining: assume for example a peer-to-peer system to be designed over a network consisting of several smaller networks linked through few gateways. Another example is when the underlay relies on a social network, that is the underlay only connects peers sharing some area of interest.

### 3.3 Global Objectives for Overlays

Despite its natural distributed construction, the overlay graph is here seen as a single object. This is in line with users' perception of the overlay and the services it provides. We aim now to define an objective metric that could characterize the performance of a particular configuration $C$. Formally, we try and address the problem of defining a function $f$ that returns, given a configuration $C$, a value from an ordered domain. We are interested in configurations that maximize the output of this function. In the following we discuss various functions and we arbitrarily choose one of them for the sequel of this paper.

**Maximizing Resource Sharing:** the most natural approach to measure the optimality of a configuration is to simply account for the total amount of resources received by all peers. Hence we define a function $f_1$ that measures $\sum_{i \in V} \sum_{j \in V} w(\overrightarrow{ji})$. This function captures the ability of each peer to receive a given service from the overlay. The optimal value of $f_1$ is easy to determine: when a feasible configuration $C$ verifies that the total amount of resources owned by the peers is comparable to the total amount of exchanged resources, $f_1(C)$ is obviously equal to $R$. The task of the overlay designer is then to conceive a configuration that optimally exploits the available resources if such configuration exists.

The problem of $f_1$ is that it does not account for the final resource distribution: some peers may be very well served while some others may not receive resources at all. This issue can sometimes be neglected, for example when a common authority such as an operator is in control of all peers. However, when peers are controlled by un-coordinated entities seeking to maximize their benefit from the system, then $f_1$ would result in some peers being better-off by disconnecting from the overlay, thus diminishing the amount of resources available to the system. This could result in a vicious circle and end up in the service disruption.

**Maximizing Peer Utilities:** if peers can are assumed to be rational, selfish agents, it is natural to measure optimality accounting for the ratio between the resources a peer receives and the resources it provides to the system. For a peer $i$, we define this ratio as: $\alpha(i) = \frac{\sum_{j \in V} w(\overrightarrow{ji})}{\mathcal{R}(i)}$. Hence, we term $f_2$ the function that measures the system performance and which is defined as $f_2 = \sum_{i \in V} \alpha(i)$. Unfortunately, $f_2$ suffers from the same problems that affect $f_1$ as it is unfair with respect to peers with disparate contribution levels.

**Introducing Fairness:** a traditional way to introduce the notion of *fairness* into a system is to define the thresholds $\alpha_{min}$ and $\alpha_{max}$ which are respectively the minimal and maximum contribution ratio that globally characterize the system. We thus define $f_3 = \frac{1}{1+(\alpha_{max}-\alpha_{min})}$ which is maximized when the difference between $\alpha_{min}$ and $\alpha_{max}$ is minimal. Here, the best configuration is considered as the one guarantying $\alpha_{max} = \alpha_{min}$.

In the sequel of this paper, we focus on perfectly reciprocal overlays, $i.e$ configurations that maximize the quantity $\alpha_{min}$, which we term $\alpha$ for simplicity. We assume self-sustainable peer-to-peer systems, so no exogenous resources can be injected in the system: the case for $\alpha > 1$ is impossible in our setting. Interestingly, we observe that maximizing $\alpha$ (*i.e.* is equal to one) also maximizes the values of both functions $f_1$ and $f_3$. Indeed, if $\alpha = 1$, then, for every peer $i$, we have that $\sum_{j \in V} w(\overrightarrow{ji}) = \mathcal{R}(i)$, which gives $\sum_{i \in V} \sum_{j \in V} w(\overrightarrow{ji}) = R$ and $\alpha_{max} = \alpha_{min}$. Hence, both $f_1$ and $f_3$ reach their global maximum. Note also that if $\alpha = 1$, then the value of $f_2$ is exactly the number of peers of the system.

# 4 Results for Perfectly Reciprocal Overlays

In this section, we study possible solutions to the resource sharing problems defined in Sec. 3. In the following we assume global knowledge of the underlay graph: the study of *distributed* algorithms to achieve the following solutions is outside the scope of this paper and is subject of future research. For the sequel, we identify the peers of the overlay graph with $x_1, \cdots, x_n$ and with $e_1, \cdots, e_m$ the set of overlay edges $\overleftrightarrow{E_U}$.

## 4.1 Basic Results

**Known Overlay Graph, Knwon Resource Distribution:** In order to solve this problem, it is sufficient to solve a system of linear equations with $2n$ equations with $m$ variables. The unknowns of the system are the weights of the overlay graph: $w(e_i)$ with $i \in \{1, \cdots, m\}$. For each peer $x_i$, we observe that the sum of the resources that this peer share with other peers is equal ot $\mathcal{R}(x_i)$. Thus, for peer $x_i$, we the following equation holds: $\sum_{x_j \in N_U^+(x_i)} w(\overrightarrow{x_i x_j}) = \mathcal{R}(x_i)$. Given the definition of optimality discussed in Sec. 3.3, our goal is to achieve $\alpha = 1$: this means that the sum of resources that the peer $x_i$ receives from other peers it is connected to is also $\mathcal{R}(x_i)$. For $x_i$ it holds that: $\sum_{x_j \in N_U^-(x_i)} w(\overrightarrow{x_i x_j}) = \mathcal{R}(x_i)$. Summing over all $n$ peers of the overlay we have a system of $2n$ linear equations with $m$ variables. If this system is solvable, then each of its solution provides the assignment of weights such that $\alpha = 1$. This computation can be done in $\mathcal{O}(n.m^2)$ with the Gauss-Jordan elimination process.

**Known Overlay Graph, Unknown Resource distribution:** In this case, the existence of an assignment of weights $w(e_i)$ and of resources $\mathcal{R}(x_i)$ for a given peer $x_i$ such that the optimality condition $\alpha = 1$ is achieved is given by the existence of a solution to a system of $2n$ linear equations with $m + n$ variables. As derived for the previous problem, the same set of equations holds, where the difference is that the $\mathcal{R}(x_i)$ are now unknowns too. As the complexity of the Gauss-Jordan algorithm is $\mathcal{O}(k.l^2)$ where $k$ is the number of equations and $l$ the number of variables of the system, determining the existence of a solution to the system of linear equation and the computation of a solution, if one exists, can be done in $\mathcal{O}(n^3.m^2)$.

**Unknown Overlay Graph, Unknown Resource Distribution:** This problem appears to be as hard to solve as the two previous cases. At a first glance, we could use the same linear system of equations previously derived with $\mathcal{R}(x_i)$, $w(\overrightarrow{e_i})$'s and $\overrightarrow{e_i} \in \overleftrightarrow{E}$ as unknowns. However, no isolated peer exists in our system definition (that is, every peer has at least one physical connection to another peer): hence, there always exists a resource $\mathcal{R}(x_i)$ and weight $w(e_i)$ assignment such that $\alpha = 1$.

In practice this means that we always can find a solution when the overlay graph and the resource distribution are not fixed a-priori. Here we give a possible solution to the problem. For each peer $x_i$, we let $\mathcal{R}(x_i) = |N(x_i)|$, where $|N(x_i)|$ is the size of peer $x_i$'s neighborhood in the underlay graph. In order for the optimality condition $\alpha = 1$ to hold, it is sufficient to take assign $w(\overrightarrow{e}) = 1$ for each $\overrightarrow{e} \in \overleftrightarrow{E}$. Hence, for every peer $x$, we obtain $\sum_{y \in N_U^+(x)} w(\overrightarrow{xy}) = \sum_{y \in N_U^-(x)} w(\overrightarrow{yx}) = |N(x)| = \mathcal{R}(x)$, and thus we have $\alpha = 1$.

## 4.2 Polynomial-Time Decision Algorithm

We discuss here the most difficult problem that deals with the construction of an optimal overlay, for which the condition $\alpha = 1$ holds, given a resource distribution $\mathcal{R}(x_i)$. This problem can be addressed under different angles. The predominant approach in the literature is to consider a selfish setting in which every peer determines the best "rewiring" strategy to optimize some local utility function. Initial efforts, pioneered by [6], define a *network creation game* in which nodes have to establish links to distant nodes at a minimum cost while maximizing a distance-based quality of service metrics and prove the NP-hardness of the problem. Others [11], define the *selfish neighbor selection* problem and provide some heuristics based on the $k$-median problem on asymmetric distance when the out-degree of a peer (that is the number of connections it can establish is bounded). We discuss more on the particular case of bounded out-degree in Sec. 4.3.

In this paper we make the following observation: the hard problem of establishing an optimal overlay can be re-conducted to the max-flow problem. In a max-flow problem, the goal is to find the maximal value that a flow between a single source and a single sink can achieve in a network where each edge $\overrightarrow{xy}$ has a maximal nominal capacity $c(\overrightarrow{xy})$. This problem has been widely studied in the literature and the two most known algorithms that achieve an optimal solution are the Ford-Fulkerson [7] and the Edmonds-Karp [5] algorithms. These algorithms respectively have a time complexity in $\mathcal{O}(m.f)$ and $\mathcal{O}(n.m^2)$ where $n$ is the number of vertices of the flow network, $m$ the number of edges and $f$ the value of the maximal flow.

We now give some definitions that allow to transform the problem of the overlay creation into a max-flow problem. We define the network $N = (V', E', c)$ associated to our problem. We define the sets of vertices of $N$ as the set containing a sink, a source and for every peer $x$, we replace it by the vertices $x^+$ and $x^-$. Formally, we have $V' = \{s, t\} \cup \{x_i^+, x_i^- : i \in \{1, \cdots, n\}\}$, where the node $s$ is the source and the node $t$ is the sink of the network. Now, we create oriented edges from the source to each vertex $x_i^+$ and edges from each vertex $x_i^-$ to the sink. Moreoever, if there is an edge $\{x, y\}$ in the underlay graph, we assign one edge from $x^+$ to $y^-$ and one edge from $y^+$ to $x^-$. Thus we can define $E'$ as $E' = \{\overrightarrow{sx_i^+}, \overrightarrow{x_i^- t} : i \in \{1, \ldots n\}\} \cup \{\overrightarrow{x_i^+ x_j^-}, \overrightarrow{x_j^+ x_i^-} : \{x_i, x_j\} \in E\}$. It remains, to define the capacity of each edge of the network. For every edge $\overrightarrow{sx_i^+}$ or $\overrightarrow{x_i^- t}$, we define its nominal capacity as $\mathcal{R}(x_i)$ while for all other edges the capacity is unbounded (as shown later, we also can assign a capacity equal to $R$). The following proposition holds:

**Proposition 1** *The value of the maximal flow of $N$ is $R$ iff the maximal value of $\alpha$ is* 1.

*Proof.* For the forward implication, let $f$ be a flow achieving the maximal value $R$. By the definition of $R$ and of the capacity of edges, for every $i \in \{1, \cdots, n\}$ we obtain that $f(\overrightarrow{sx_i^+}) = f(\overrightarrow{x_i^- t}) = \mathcal{R}(x_i)$. It is now sufficient to take $w(\overrightarrow{x_i x_j}) = f(\overrightarrow{x_i^+ x_j^-})$. Indeed, due to the flow conservation principle, for every peer $x_i$, we directly obtain $\sum_{x_j \in N_U^+(x_i)} w(\overrightarrow{x_i x_j}) = \sum_{x_j \in N_U^-(x_i)} w(\overrightarrow{x_j x_i}) = \mathcal{R}(x_i)$, and thus $\alpha = 1$.

For the backward implication, assume that $\alpha = 1$ and that we have an assignment of the weights which reaches this bound. Now we define a flow $f$ for $N$ as follows: for every vertex $x_i$, let $f(\overrightarrow{sx_i^+}) = f(\overrightarrow{x_i^- t}) = \mathcal{R}(x_i)$ and for every distinct vertex $x_i$, $x_j$, let $f(\overrightarrow{x_i^+ x_j^-}) = w(\overrightarrow{x_i x_j})$. Now as $\alpha = 1$, we have that for every peer $x_i$, $\sum_{x_j \in V} w(\overrightarrow{x_i x_j}) = \sum_{x_j \in V} w(\overrightarrow{x_j x_i}) = \mathcal{R}(x_i)$, and thus we have that $f$ is a well defined flow. Now, as the only neighbors of the source are the $x_i^+$'s and as the only neighbors of the sink are the $x_i^-$'s and by the definition of the capacity of this edges, it is clear that the value of $f$ is a maximal one. □

With this proposition, we state that it is possible to determine if an assignment of weights such that $\alpha = 1$ exists. Moreover, as seen in the proof, in the same time we can determine such an assignment, if it exists. If we use the Ford-Fulkerson algorithm, we obtain a time complexity in $\mathcal{O}((n + |E|).R)$ where $E$ is the number of edges of the underlay graph and if we use the Edmonds-Karp algorithm, the complexity is in $\mathcal{O}(n.(n + |E|)^2)$. Figures 4 and 5 provide an intuitive illustration of our methodology. Fig. 4*a* is the input to our problem: an underlay graph with a given resource distribution $\mathcal{R}(x_i)$ for every peer $x_i \in V$. Our goal is to obtain an optimal overlay network which, for every peer $x_i \in V$, assigns the available resources to some selected neighbors. Fig. 4*b* represents our first step, in which we proceed with the reduction to a flow network. In Fig. 5*a* we show the max-flow network and in Fig. 5*b* we illustrate the resulting overlay network which optimally assigns the available resources at every peer such as $\alpha = 1$.

### 4.3 NP-Completeness for Bounded Outdegree Overlays

In this Section we assume a peer $x_i \in V$ to have not more than $d(x_i)$ connections, that is, the overlay graph has a bounded out-degree: $|N_U^+(x_i)| \leq d(x_i)$. This assumption is not only helpful to reduce the search space for the design of an optimal overlay graph, given a resource distribution $\mathcal{R}(x_i)$, but stems from practical considerations. Indeed, in most real systems the number of connections a peer can establish concurrently is limited, both for scalability and performance reasons.

As for the general case (where we have no constraints on outbound connections), we reduce the problem of the overlay construction to a max-flow problem: the constraint on the number of connections a peer can establish becomes a constraint on the out-degree of the network flow. Hence, the problem is to determine the maximum flow of a *edge-capacitated* digraph under the constraint that for each vertex $x_i \in V$, the out-degree induced by the flow is less or equal to $d(x_i)$. Formally, we express our problem as follows:

**Bounded-outdegree Maximal flow**

INSTANCE : Directed graph $G = (V, A)$, specified verticves $s$ and $t$, capacity $c(a) \in \mathbb{N}^*$ for $a \in A$, bound on out-degree $d(v) \in \mathbb{N}^*$ for $v \in V$ and positive integer $K$.

QUESTION : Is there a flow function $f : A \longrightarrow \mathbb{N}$ such that :

*(i)* $f(a) \leq c(a)$ for all $a \in A$,

*(ii)* for each $v \in V \setminus \{s,t\}$, $\sum_{(u,v)\in A} f((u,v)) = \sum_{(v,u)\in A} f((v,u))$, *i.e.* flow is *conserved* at $v$,

*(iii)* for each $v \in v$, $|\{u \in V : f(v,u) \neq 0\}| \leq d(v)$, *i.e.* the outdegree of $v$ in the flow is at most $d(v)$,

*(iv)* $\sum_{(u,t)\in A} f((u,t)) - \sum_{(t,u)\in A} f((t,u)) \geq K$, *i.e.* the net flow in $t$ is at least $K$.

First, let's recall the definition of the minimum cover problem, which is NP-complete and then we prove here that our problem is NP-complete even if the digraph induced by $V \setminus \{s, v\}$ is bipartite.

**Minimum cover**

INSTANCE : Collection $C$ of subsets of a finite set $S$, positive integer $L \leq |C|$.

QUESTION : Does $C$ contain a cover for $S$ of size $L$ or less, *i.e.* a subset $C' \subseteq C$ with $|C'| \leq L$ such that every element of $S$ belongs to at least one member of $C'$ ?

**Theorem 1** *Bounded-outdegree Maximal flow is NP-complete.*

*Proof.* Given an instance of Bounded-outdegree Maximal flow and a flow $f$, verifying the conditions $(i)$ to $(iv)$ is polynomial in the size of the problem: hence, Bounded-outdegree Maximal flow belongs to NP.

We now transform Minimum Cover to Bounded-outdegree Maximal flow. Let $C = \{C_1, \cdots, C_m\}$ a collection of subsets of a finite set $S = \{x_1, \cdots, x_n\}$ and let $L$ a positive integer such that $L \leq |C|$. We define $G = (V, A)$ with $V = \{s,t\} \cup S \cup C$ and $A = \{(s, C_i) : i \in \{1, \cdots, m\}\} \cup \{(x_i, t) : i \in \{1, \cdots, n\}\} \cup \{(C_i, x_j) : i \in \{1, \cdots, m\}, j \in \{1, \cdots, n\}$ and $x_j \in C_i\}$. Let $c((s, C_i)) = |C_i|$ and $c(a) = 1$ otherwise and let $d(s) = L$ and $d(v) = |V|$ otherwise and let $K = |S|$ It is clear that the instance of Bounded-outdegree Maximal flow can be constructed in polynomial time. We claim that $C$ contains a cover for $S$ of size $L$ or less if and only if there exists a flow $f$ which respect the conditions $(i)$ to $(iv)$.

Clearly, if we have a minimum cover $C'$ of $C$ of size lower or equal to $L$, then we can define a mapping $\phi$ from $S$ to $C'$ such that $x_i \in \phi(x_i)$. Then we can define the flow $f$ as $f((s, C_i) = 0$ if $C_i \notin C'$ and $f((s, C_i) = |\phi^{-1}(C_i)|$ otherwise, $f((C_i, x_j)) = 1$ if $C_i = \phi(x_j)$ and 0 otherwise, and $f((x_j, t)) = 1$. One can check that $f$ verifies conditions $(i)$ to $(iv)$ of the definition of Bounded-outdegree Maximal flow.

Now, let $f$ a flow that fulfills conditions $(i)$ to $(iv)$ of the definition of Bounded-outdegree Maximal flow. One can check that $C' = \{C_i : f((s, C_i)) \neq 0\}$ is a minimum cover of $S$ of size lower or equal to $L$. Indeed, as $K = |S|$, the definition of the $G$ implies that for every $x_j \in S$, there exists $C_i \in C'$ such that $x_j \in C_i$ (we must have that $f((C_i, x_j)) = 1$). Now as $d(s) \leq L$, we have that $|C'| \leq L$. □

We have proved that Bounded-outdegree Maximal flow is NP-complete even if the digraph is bipartite. Since our original problem is to determine if there exists an overlay graph such that $\alpha = 1$ when the digraph is bipartite, then this is also a NP-complete problem.

# 5 Case Study: *b*-Matching-Based Overlays

In this Section, we describe a typical application of the theoretical framework: studying the optimality of a distributed algorithm for overlay construction. The object of this study is the *b*-matching theory.

## 5.1 Non-Optimality of the *b*-Matching Theory

Stable matching theory (in particular stable roommates problems) provides a fertile ground to devise distributed algorithms used in peer-to-peer networks. Here we focus on a particular instance of stable-matching theory, namely *b*-matching. This model has recently provided insights into peer download rates and a good understanding of upload setting strategy in peer-to-peer content distribution applications [8, 9, 12]. In such systems and under some constraints, it has be shown that there exists an unique stable solution matching peers exhibiting similar traits in terms of up/downlink bandwidth which is optimal and a fast decentralized algorithm has been designed to reach this stable solution. Given an "acceptance" graph $G = (V, E)$ (which corresponds here to the underlay graph), each peer $x$ ranks its neighbors in the acceptance graph according to a preference list. Consider for example a peer $x$ having two neighbors $u$ and $v$. If $u$ exhibits, for the peer $x$, a better ranking than $v$, then $x$ would prefer to link with $u$ rather than with $v$. We focus here on a global ranking, *i.e.* each vertex $i$ has a global utility $S(i)$ which induces a total order on peers. The goal for each peer it to maximize its local payoff by being matched with its best neighbors with respect to global utility values when the degree of a peer $x$ is bounded by $b(x)$.

The *b*-matching theory is claimed to build efficient overlays, so is an excellent target for a study using the proposed framework. Given an underlay graph $\overleftrightarrow{G}$ and a resource set $\mathcal{R} = \{\mathcal{R}(x) : \forall x \in V\}$, one can use the *b*-matching algorithm [12] in order to build an overlay graph $\overleftrightarrow{E}_U$. The total order on peers is naturally matched with their resources, that is, the utility of a peer $x$ is $S(x) = \mathcal{R}(x)$. The idea now is to check if, given $\overleftrightarrow{G}$ and $\mathcal{R}$, the overlay $\overleftrightarrow{E}_U$ can be a perfectly reciprocal overlay. An issue is to fix a value $b$ for peer degrees. We propose to act as follows. Given $\overleftrightarrow{G}$ and $\mathcal{R}$, we first compute an optimal unbounded overlay $(\overleftrightarrow{E'}_U, w)$ by using the algorithm proposed in Sec. 4. Then we measure the maximal degree in this overlay graph $\max\{|N_U^+(x) \cup N_U^-(x)| : x \in V\}$ and we fix $b$ to this value. The overlay $(\overleftrightarrow{E'}_U, w)$ can be seen as a competitor for the *b*-matching-based overlay $\overleftrightarrow{E}_U$ in the sense that it is easy to fix to 0 some links in $\overleftrightarrow{E'}_U$ in order to have exactly the same number of edges in $\overleftrightarrow{E'}_U$ than in $\overleftrightarrow{E}_U$. Note that if there is no solution for unbounded overlay graph, we do not run the *b*-matching algorithm because we only focus on checking whether the *b*-matching overlay can reach an existing optimal configuration.

As depicted in Fig. 1, a first immediate result is the non-optimality of $b$-matching overlays. Indeed, given a basic underlay and a resource set, the $b$-matching technique fails in building a perfectly reciprocal overlay although there exists an overlay which achieves this goal for the same underlay and resource set. The $b$-matching theory was, to our opinion, a good candidate for a generic distributed algorithm for building perfectly reciprocal overlays. With this theoretical non-optimality result, the problem remains open.

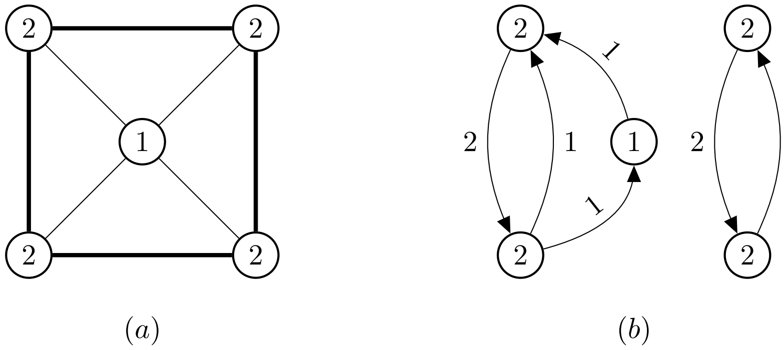


Figure 1: ($a$): An underlay graph with 5 peers where bold edges form the overlay built by the 2-matching and ($b$): a solution achieving $\alpha = 1$ such that every peer interact with at most two other peers

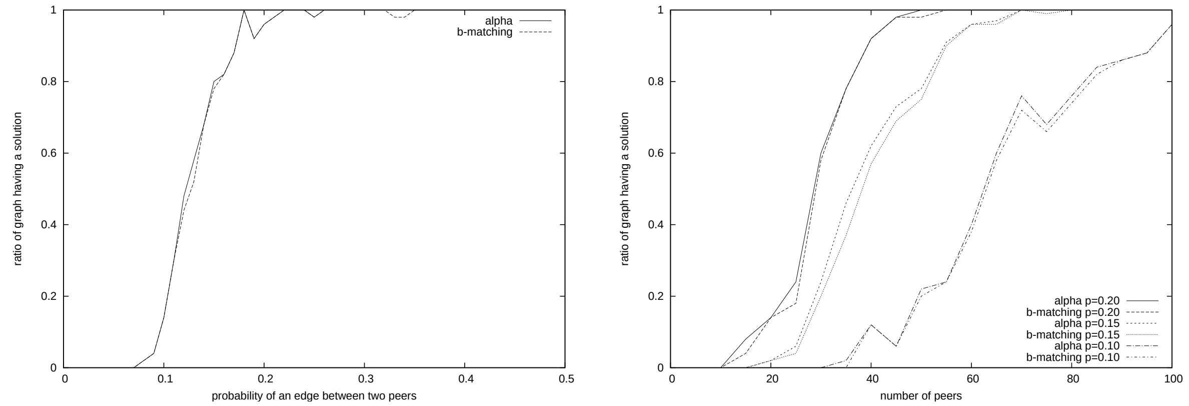
## 5.2 Efficiency Study on Random Graphs

Even if the $b$-matching techniques has been proved to not build the optimal overlay in some cases, we evaluate nevertheless its efficiency when this techniques is applied on random graphs. We aim to know how frequent does the case where the $b$-matching fails occur.

We develop a basic simulator with the following settings. Peers are labeled from 1 to $n$ where $n$ is the number of peers. We use the Erdös-Renyi loopless symmetric graphs $\mathcal{G}(n, p)$ as underlay graph, where $p$ is the probability that a given edge exists, so the expected degree $d = p(n - 1)$. Resources are uniformly distributed over peers: a generic numerical value indicating the amount of available resources is assigned in the set $[10, 100]$. To contravene random effects, each plot is obtained after 100 iterations. We count the number of underlay and resource set on top of which a perfectly reciprocal overlay can be built, then the number of times a perfectly reciprocal configuration can be created from an overlay built by the $b$-matching technique. It is easy to evaluate the efficiency of $b$-matching techniques by computing the ratio of optimal configurations that may derive from this distributed algorithm.

First, we analyze the impact of the number of edges. We intuitively expect that an almost empty graph (with a small number of edges) does not have an optimal solution with high probability whereas a graph which is almost a clique (*i.e.* a graph where every peer is connected to each peer) has a high probability to have an optimal solution.

Fig. 2(a) shows the results obtained for a graph with $n = 50$ peers. The probability for an edge to exist goes from 0 (the empty graph) to 0.5 (expected degree is roughly 25). The first interesting result is that as soon as the expected degree is at least 10 (which corresponds to a probability of 0.2), in most cases (at least 96%) there is an optimal repartition of the resources. Obviously, given an arbitrary probability $p$ it is always possible to find a random graph $\mathcal{G}(n, p)$ and a resource distribution such that there is no optimal repartition of these resources. However, our results show that the number of graphs having no optimal overlay is very small. The second interesting (and surprising) fact is that the $b$-matching algorithm succeeds in building perfectly



(a) Comparison of the evolution of percent of graph having optimal solution and percent of graph having optimal solution with overlay fixed by the matching algorithm when $p$ increases.

(b) Evolution of percent of graph having optimal solution and percent of graph having optimal solution with overlay fixed by the matching algorithm when $p = 0.20$, $p = 0.15$ and $p = 0.10$ and when the number of peers increases.

Figure 2:

reciprocal configurations for almost all cases. So the failure case for $b$-matching techniques seem very infrequent on random graphs.

We now make vary the number of peers in an interval $[0, 100]$. Fig. 2(b) shows results for three values of $p$: $p = 0.10$, $p = 0.15$ and $p = 0.20$. The $b$-matching algorithm achieves an optimal configuration in most cases; furthermore, we observe that as the average degree of the underlay graph increases, the probability that a given configuration has an efficient resources allocation imposed by the overlay increases.

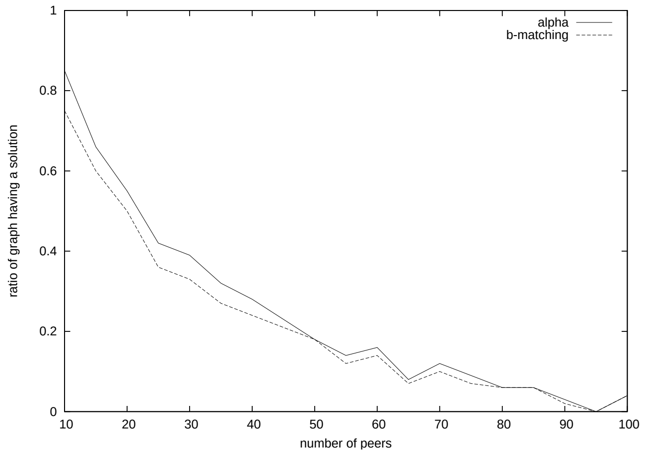


Figure 3: Evolution of percent of graph having optimal solution and percent of graph having optimal solution when the expected degree is 5 and the number of peers increases.

Our last study focuses on the case when the expected degree of the underlay graph is fixed. Fig. 3 illustrates results for $n \in [10, 100]$ when the expected degree of the underlay is bound to 5. Since the expected degree is fixed, the number of solution decreases when the number of peers increases. We note that the number of solutions given by the $b$-matching can be far from those given in the general case. Indeed, for graphs having 10 peers, 85% of underlays admit a solution but *only* 75% of them have a solution based on $b$-matching. As the number of peers increases, this difference decreases: it is null for graphs with 100 nodes.

## 6 Conclusion and future works

In this paper we have proposed a general framework for the problem of rival resources sharing in overlay networks. We have focused on perfectly reciprocal overlays, *i.e.* peer-to-peer systems where each peers receives exactly the same amount of resources it offers. Given a fixed underlay graph, the existence of such overlays is a fundamental question. We first show that it is possible to decide whether a optimal configuration exists in polynomial time. We then prove that the same problem becomes NP-complete for perfectly reciprocal bounded outdegree overlays. Finally, we have used the theoretical bound given by this framework with numerical simulations in order to show the nearly optimality of overlay networks designed by the decentralized algorithm provided by the $b$-matching model.

In our study, we have only considered perfectly reciprocal overlays but various other optimization may be considered. In particular, a new global objective for overlay could be to maximize the number of peers receiving at least $k$ resources. In this case, the optimality is achieved when all peers receive at least $k$ resources. This case makes especially sense for peer-to-peer live streaming systems where a peer can play the stream only if it is able to receive enough data, so if its overlay neighbors can deliver a sufficient amount of bandwidth resources. In another possible future work, we could go deeper in understanding perfectly reciprocal overlays. More precisely, many peer-to-peer systems admits a neighborhood size in $\mathcal{O}(\log n)$. For this family of overlays, the complexity of determining whether a perfectly reciprocal configuration exists may be not NP-complete but the problem is still open.

## References

[1] E. Altman, D. Barman, B. Tuffin, and M. Vojnovic. Parallel tcp sockets: Simple model, throughput and validation. In *Proc. of IEEE INFOCOM*, Barcelona, Spain, May 2006.

[2] E. Anshelevich, A. Dasgupta, J. Kleinberg, E. Tardos, T. Wexler, and T. Roughgarden. The Price of Stability for Network Design with Fair Cost Allocation. In *Proc. of ACM STOC*, Chicago, IL, June 2004.

[3] E. Anshelevich, A. Dasgupta, E. Tardos, and T. Wexler. Near-Optimal Network Design with Selfish Agents. In *Proc. of ACM STOC*, San Diego, CA, June 2003.

[4] Y. Benkler. *The Wealth of Networks.* Yale University Press, 2006.

[5] J. Edmonds and R. M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM*, 19(2):248–264, 1972.

[6] A. Fabrikant, A. Luthra, E. Maneva, C. H. Papadimitriou, and S. Shenker'. On a Network Creation Game. In *Proc. of ACM PODC*, Boston, Massachusetts, July 2003.

[7] L. R. Ford, Jr. and D. R. Fulkerson. Maximal flow through a network. *Canad. J. Math.*, 8:399–404, 1956.

[8] A.-T. Gai, D. Lebedev, F. Mathieu, F. de Montgolfier, J. Reynier, and L. Viennot. Acyclic preference systems in p2p networks. In *Proc. of Euro-Par Conf.*, pages 825–834, august 2007.

[9] A.-T. Gai, F. Mathieu, F. de Montgolfier, and J. Reynier. Stratification in p2p networks: Application to bittorrent. In *Proc. of IEEE ICDCS*, 2007.

[10] D. Hughes, G. Coulson, and J. Walkerdine. Free riding on gnutella revisited: The bell tolls? *IEEE Distributed Systems Online*, 6(6), 2005.

[11] N. Laoutaris, G. Smaragdakis, A. Bestavros, and J. W. Byers. Implications of Selfish Neighbor Selection in Overlay Networks. In *Proc. of IEEE INFOCOM*, Anchorage, AK, May 2007.

[12] D. Lebedev, F. Mathieu, L. Viennot, A.-T. Gai, J. Reynier, and F. de Montgolfier. On using matching theory to understand p2p network design. In *Int. Network Optimization Conf.*, pages 1–6, 2007.

[13] D. Qiu and R. Srikant. Modeling and performance analysis of bittorrent-like peer-to-peer networks. In *Proc. of ACM SIGCOMM*, August 2004.

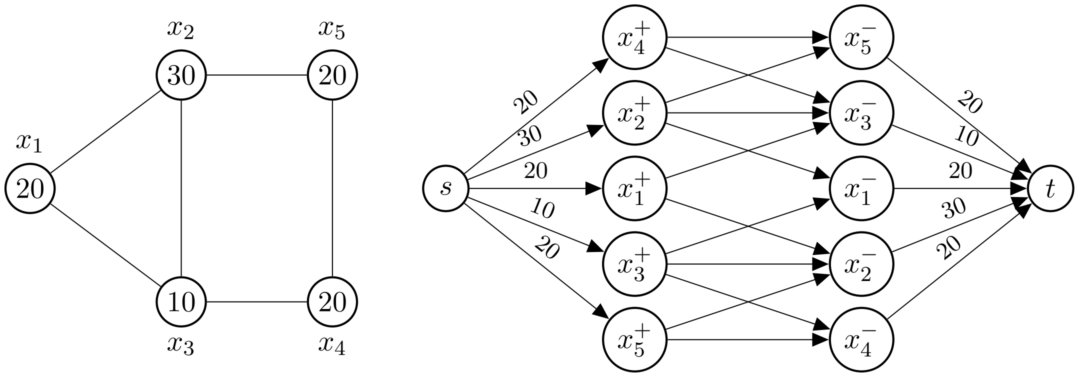
[14] L. S. Shapley. *A Value for n-person Games*. Princeton University Press.

[15] G. Smaragdakis, N. Laoutaris, P. Michiardi, A. Bestavros, J. W. Byers, and M. Roussopoulos. Swarming on Optimized Graphs for n-way Broadcast. In *Proc. of IEEE INFOCOM*, Phoenix, AZ, April 2008.

[16] D. Stutzbach, D. Zappala, and R. Rejaie. The scalability of swarming peer-to-peer content delivery. In *Proc. of IFIP Networking*, May 2005.

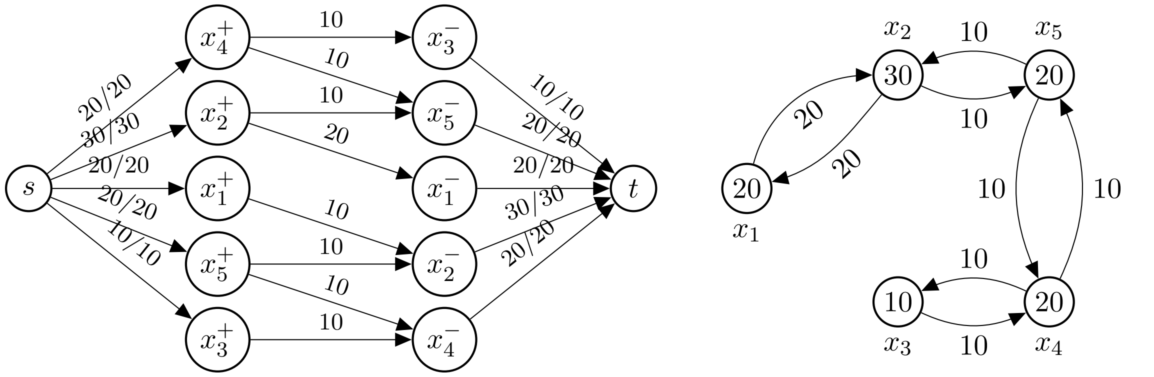
[17] W. Wang and B. Li. Market-driven bandwidth allocation in selfish overlay networks. In *Proc. of IEEE INFOCOM*, Miami, FL, March 2005.

# A Figures



$(a)$ $(b)$

Figure 4: $(a)$: An example of resource repartition on an underlay graph and $(b)$: the associated edge-capacitated network (edges with no capacity are edges with unlimited capacity)



$(a)$ $(b)$

Figure 5: $(a)$: A maximal flow for the network of the Figure 4$a$ and $(b)$ : the associated *optimal overlay graph* with resources allocation.