



**HAL**  
open science

## A Coalition Formation Game in Partition Form for Peer-to-Peer File Sharing Networks

Walid Saad, Zhu Han, Tamer Basar, Merouane Debbah, Are Hjørungnes

► **To cite this version:**

Walid Saad, Zhu Han, Tamer Basar, Merouane Debbah, Are Hjørungnes. A Coalition Formation Game in Partition Form for Peer-to-Peer File Sharing Networks. IEEE GLOBECOM 2010, Dec 2010, United States. 5 p. hal-00556161

**HAL Id: hal-00556161**

**<https://centralesupelec.hal.science/hal-00556161>**

Submitted on 15 Jan 2011

**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.

# A Coalition Formation Game in Partition Form for Peer-to-Peer File Sharing Networks

Walid Saad<sup>1</sup>, Zhu Han<sup>2</sup>, Tamer Başar<sup>3</sup>, Merouane Debbah<sup>4</sup>, and Are Hjørungnes<sup>1</sup>

<sup>1</sup>UNIK - University Graduate Center, University of Oslo, Kjeller, Norway, Email: {saad, arehj}@unik.no

<sup>2</sup>Electrical and Computer Engineering Department, University of Houston, Houston, USA, Email: zhan2@mail.uh.edu

<sup>3</sup>Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, USA, Email: basar1@illinois.edu

<sup>4</sup>Alcatel-Lucent Chair in Flexible Radio, SUPÉLEC, Gif-sur-Yvette, France, Email: merouane.debbah@supélec.fr

**Abstract**—In current peer-to-peer file sharing networks, a large number of peers with heterogeneous connections simultaneously seek to download resources, e.g., files or file fragments, from a common seed at the time these resources become available, which incurs high download delays on the different peers. Unlike existing literature which mainly focused on cooperative strategies for data exchange between different peers *after* all the peers have already acquired their resources, in this paper, we study the cooperation possibilities among a number of peers seeking to download, concurrently, a number of resources *at the time the availability of the resources is initially announced at a common seed*. We model the problem as a coalitional game in partition form and we propose an algorithm for coalition formation among the peers. The proposed algorithm enables the peers to take autonomous decisions to join or leave a coalition while minimizing their average download delay. We show that, by using the proposed algorithm, a Nash-stable partition composed of coalitions of peers is formed. Within every coalition, the peers distribute their download requests between the seed and the cooperating partners in a way to minimize the total average delay incurred on the coalition. Analytically, we study the 2-peer scenario and derive the optimal download request distribution policies. Simulation results show that, using the proposed coalition formation game, the peers can improve their average download delay per peer of up to 99.6% compared to the non-cooperative approach for the case with  $N = 15$  peers.

## I. INTRODUCTION

Peer-to-peer (P2P) file sharing services such as BitTorrent [1] have recently become one of the most popular resource sharing medium among users. Using a combination of advanced search and communication techniques as well as large-scale distributed file storage systems, P2P file sharing systems allow a number of users to download and share content among each other in a decentralized, scalable, and fault tolerant manner.

Due to the heterogeneous capabilities of the peers as well as the decentralized nature of P2P file sharing, current P2P systems face many challenges at different perspectives [2]. For instance, in [3], the authors proposed incentives, using tit-for-tat strategies, for allowing the peers in a P2P system to efficiently exchange their data. The work in [4] studied the problem of resource reciprocation of multimedia content among peers using a stochastic game. Further, in [5], mesh-based architectures for content dissemination are studied in P2P networks.

Most of the aforementioned work in P2P networks focuses on the P2P sharing of resources *after* these resources were *entirely* acquired by the peers, independently, from different seeds. However, a prominent challenge of P2P systems that remains relatively unexplored is the study of the scenario where a number of peers compete for downloading, concurrently, a number of resources, e.g., files or file fragments that complement their already owned resources, from the *same seed at the time the availability of the resources is announced*. For instance, whenever a new file, fragment, or group of files/file fragments, become available at a certain seed, i.e., a server or another peer, a large number of interested peers, e.g., *a flash crowd* of nodes interested

This work was supported by NFR through the project 183311/S10, and US NSF CNS-0910461, CNS-0953377, and CNS-0905556.

in these resources, will, concurrently, attempt to download these resources from the seed. Eventually, due to the limited upload capabilities of the seed and the heterogeneous characteristics of the peers, this scenario yields increased download delays on the competing peers (this issue is highlighted in [5]). Hence, one important challenge is to propose cooperative schemes for P2P networks that can alleviate the download delays of the peers during this phase of simultaneous download of resources at the time they become available.

The main contribution of this paper is to propose cooperative strategies, using the framework of coalition formation games in partition form, for enabling the peers to improve their download delays during the concurrent download of resources from a common seed. Another contribution of this paper is to discuss the optimal division of the peers' download request between the main seed and other cooperative peers. For this purpose, we formulate a coalition formation game in partition form among a number of peers that seek to cooperatively improve their download delay while downloading resources from a common seed at the time the resources become available. For the devised game model, we propose a coalition formation algorithm through which each peer can take an individual decision to join or leave a coalition while minimizing its average download delay. We show that the proposed coalition formation algorithm enables the peers to self-organize into a Nash-stable partition composed of coalitions of peers. Analytically and through simulations, we assess different aspects of the proposed scheme.

## II. SYSTEM MODEL

Consider a P2P network consisting of  $N$  peers and let  $\mathcal{N}$  denote the set of all peers. We consider that each peer already owns a small number of resources, e.g., files or fragments, related to a particular content and is seeking to download the remainder of this content. Whenever a seed, e.g., a server or another independent peer (not in the set  $\mathcal{N}$ ) that owns the entire content of interest to the peers, announces the availability of a number of resources, all  $N$  peers will concurrently attempt to download these resources from the seed. In this model, it is considered that the seed owns *all* the data related to the content of common interest to the peers while each peer in  $\mathcal{N}$  owns different chunks of this data.

Each peer  $i \in \mathcal{N}$  has an upload rate  $\mu_i$ , and, in a non-cooperative manner, attempts to download the remainder of its data from the common seed with a download rate  $\lambda_i$ . With no loss of generality, we assume that, for any peer  $i$ , the arrival process for the download requests is exponentially distributed with parameter  $\lambda_i^{-1}$ . In such a scenario, given that the seed has an upload rate of  $\mu^s$  and considering that the file service process is an M/D/1 queue, the average delay  $\tau_{nc}$  for any peer  $i \in \mathcal{N}$  non-cooperatively downloading data from the seed is

$$\tau_{nc} = \frac{\lambda^s}{2\mu^s(\mu^s - \lambda^s)}, \quad (1)$$

where  $\lambda^s = \sum_{i \in \mathcal{N}} \lambda_i$  is the total request rates at the seed.

As the seed has a limited upload rate  $\mu^s$ , one can see from (1) that, whenever the number of peers interested in the seed's content is large, the download delay increases significantly. In particular, the total download requests that the seed can handle must satisfy the stability condition of the queueing system in (1), i.e.,  $\lambda^s < \mu^s$ . As a direct result of this condition, the number of peers that can concurrently download the data from the seed, non-cooperatively, is limited and strongly depends on the heterogeneous download rates of the peers. For example, a single peer with large download rate can congest the seed and forbid other peers from utilizing the resource.

To reduce their performance, instead of solely downloading the remaining content from the seed directly, the peers can download the content of interest from the seed as well as from a subset of other peers. By doing so, the peers can potentially reduce the load on the seed and possibly improve their delay. Within each group of cooperating peers, i.e., a coalition  $S \subseteq \mathcal{N}$ , every peer  $i \in S$  can direct its download requests to the seed as well as to other cooperative peers  $j \in S \setminus \{i\}$  with a certain fraction  $p_{ij}$  such that

$$0 \leq p_{ij} \leq \frac{\mu_j}{\lambda_i}, \forall j \in S \setminus \{i\} \text{ and } \sum_{j \in S \setminus \{i\}} p_{ij} + p_{i0} = 1 \quad (2)$$

where the upper bound in the first condition ensures that no peer  $i \in S$  will download data from another peer  $j \in S$  with a rate that exceeds the upload rate  $\mu_j$  of peer  $j$  and  $0 \leq p_{i0} \leq \frac{\mu^s}{\lambda_i}$  is the fraction of requests directed to the seed from peer  $i$ . For every peer  $i \in S$ , we define the  $(|S| + 1) \times 1$  ( $|\cdot|$  is the set cardinality operator) column vector  $\mathbf{p}_i = [p_{i0}, \dots, p_{i|S|}]^T$  of all the fractions of requests directed by peer  $i$  to its partners in  $S$  and the seed. Note that,  $p_{ii} = 0, \forall i \in S$ . Further, we let  $\mathbf{P}_S = [\mathbf{p}_1, \dots, \mathbf{p}_{|S|}]$  be the  $(|S| + 1) \times |S|$  matrix of all the fractions of the peers in coalition  $S$ . Note that, we assume that any peer outside a coalition  $S$ , i.e., in  $\mathcal{N} \setminus S$ , will not allow peers inside  $S$  to use its upload bandwidth.

Subsequently, for any cooperative peer  $i$  member of a coalition  $S \subseteq \mathcal{N}$  which is downloading data from the seed and the peers in  $\mathcal{K} \subseteq S$  with  $|\mathcal{K}| = K$ , given that each of the  $K$  queues are independent and by using Little's Law [6] the average download delay will be

$$\tau_i(\mathbf{P}_S) = \frac{1}{K+1} \left( \sum_{j \in S, j \neq i} \frac{\Lambda_j}{2\mu_j(\mu_j - \Lambda_j)} + \tau_s \right), \quad (3)$$

where the first term is the delay from the data downloaded by peer  $i$  from  $K$  partners in coalition  $S$  with  $\mu_j$  the service rate of peer  $j \in S$ , and  $\tau_s$  the delay from the data downloaded by  $i$  from the seed which is the same for all  $i \in S$  and is given using the same expression as (1) with  $\lambda^s = \sum_{i \in S} p_{i0} \lambda_i$ . Further,  $\Lambda_j = (\mathbf{P}_S)_j \boldsymbol{\lambda}$  in (3) represents the total load at peer  $j \in S$  with  $(\mathbf{P}_S)_j$  the  $j$ th line of  $\mathbf{P}_S$  and  $\boldsymbol{\lambda}$  an  $|S| \times 1$  vector with each element  $\lambda_k$  corresponding to the download rate of a peer  $k \in S$ . Note that, we assume that, for any coalition  $S$  that forms,  $\exists i \in S$  such that  $p_{i0} \neq 0$ , i.e., *at least one peer downloads from the seed*, otherwise, the download delay in (3) is considered to be infinite. By adequately selecting the distribution of their download requests, i.e., the matrix  $\mathbf{P}_S$ , the cooperative peers might be able to improve their average.

Although the peers can divide their download requests in any way between the cooperative peers and the seed, one scheme that the peers can adopt is to distribute their requests in a way

to minimize the *total* average delay experienced by the coalition as a whole. Such an optimization is quite popular both in game theory and networking whereby a group of nodes can act so as to minimize the total social cost (or alternatively maximize the overall social welfare) on the network as a whole [7], [8].

In this context, given a coalition  $S \subseteq \mathcal{N}$  of cooperative peers, the peers in the coalition distribute their download requests, i.e., compute the vectors  $\mathbf{p}_i, \forall i \in S$  such that the total social cost, i.e., the total average delay of the coalition, is minimized. In other words, each coalition jointly solves the following optimization

$$\min_{\mathbf{P}_S} \sum_{i \in S} \tau_i(\mathbf{P}_S), \quad (4)$$

$$\text{s.t. } 0 \leq p_{i0} < \frac{\mu^s}{\lambda_i} \text{ and } 0 \leq p_{ij} < \frac{\mu_j}{\lambda_i}, \forall j \in S \setminus \{i\}, i = 1 \dots |S|,$$

$$\sum_{j \in S \setminus \{i\}} p_{ij} + p_{i0} = 1, i = 1 \dots |S|,$$

where  $\tau_i$  is given by (3) and the two constraint describe previously mentioned properties of the download request fractions. Note that, as previously mentioned, at least one peer in the coalition must be connected to the seed, i.e.,  $\exists i \in S$  with  $p_{i0} \neq 0$ , otherwise the delay is infinite. Subsequently, whenever a group of peers cooperate within a coalition  $S$ , they can compute the download request distribution which minimizes the total average delay of their coalition as per (4) which is a standard constrained optimization problem that can be solved using several known optimization methods [9]. As a result of this optimization, the delay of every peer  $i$  in coalition  $S$  will be given by  $\tau_i(\mathbf{P}_S^*)$  as per (3) with  $\mathbf{P}_S^*$  denoting the solution of (4).

Nonetheless, although minimizing the social cost is an attractive approach, in several scenarios, the social optimum and the individual incentives of the peers might not be aligned, i.e., optimizing the social cost of a coalition does not guarantee a better individual delay for every peer involved in the coalition [7]. Hence, there is a need to devise cooperative strategies which allow each peer to autonomously form coalitions while taking into account two conflicting objectives: (i)- Improving its individual delay through cooperation and (ii)- Distributing its download requests inside the coalition in a way to minimize the overall social cost on the network as per (4).

### III. PEERS COOPERATION AS A COALITIONAL GAME IN PARTITION FORM

We use the framework of coalitional game theory [8] to study the proposed cooperative P2P model. Denoting by  $\mathfrak{B}$  as the set of all partitions of  $\mathcal{N}$ , the proposed P2P cooperation problem is modeled as a coalitional game in *partition form* with *non-transferable utility* defined as follows [8]:

**Definition 1:** A coalitional game in *partition form* with *non-transferable utility* (NTU) is defined by a pair  $(\mathcal{N}, V)$  where  $\mathcal{N}$  is the set of players and  $V$  is a mapping such that for every partition  $\Pi \in \mathfrak{B}$ , and every coalition  $S \subseteq \mathcal{N}$ ,  $S \in \Pi$ ,  $V(S, \Pi)$  is a closed convex subset of  $\mathbb{R}^S$  that contains the payoff vectors that players in  $S$  can achieve.

Denoting by  $\phi_i(S, \Pi)$  the payoff of any peer  $i$  in coalition  $S \in \Pi$  within a partition  $\Pi \in \mathfrak{B}$  of  $\mathcal{N}$ , we define the coalitional value set, i.e., the mapping  $V$  in partition form, as follows

$$V(S, \Pi) = \{\phi(S, \Pi) \in \mathbb{R}^S | \forall i \in S, \phi_i(S, \Pi) = -\tau_i(\mathbf{P}_S^*)\}, \quad (5)$$

where  $\tau_i(\mathbf{P}_S^*)$  is given by (3) with  $\mathbf{P}_S^*$  the solution of (4) for coalition  $S$  and the minus sign is inserted for convenience, in

order to turn the problem into a maximization problem. Note that, the delay  $\tau_i(\mathbf{P}_S)$  in (3) depends, through its second term, not only on the download request distribution inside  $S$  but also on the distribution outside  $S$  through the seed, and, thus, it depends on the partition  $\Pi$  (for notational convenience this dependence is dropped from  $\tau_i(\mathbf{P}_S^*)$ ). Thus, the proposed P2P cooperative approach is formulated as an  $(\mathcal{N}, V)$  coalitional game in partition form with NTU where the mapping  $V$  is a closed and convex subset of  $\mathbb{R}^S$  as given by (5) (singleton set).

The main challenge of a coalitional game in partition form is to construct algorithms for forming the coalitions. First, we define the following concept:

**Definition 2:** Given any peer  $i \in \mathcal{N}$ , a *preference relation* denoted by  $\succeq_i$  is defined as a complete, reflexive, and transitive binary relation over the set of all coalition/partition pairs that peer  $i$  can be a member of, i.e., the set  $\{(S_k, \Pi) | S_k \subseteq \mathcal{N}, i \in S_k, S_k \in \Pi, \Pi \in \mathfrak{B}\}$ .

As a result, for any peer  $i \in \mathcal{N}$  that can provide a quantification of its benefit for every potential coalition that it can form, the preference relation  $\succeq_i$  can be used to compare the peer's preference between any two coalitions  $S_1 \subseteq \mathcal{N}$ ,  $S_1 \in \Pi$  and  $S_2 \subseteq \mathcal{N}$ ,  $S_2 \in \Pi'$  such that  $i \in S_1$  and  $i \in S_2$  and their respective partitions. Hence,  $(S_1, \Pi) \succeq_i (S_2, \Pi')$  implies that peer  $i$  is better off working cooperatively in coalition  $S_1$  when  $\Pi$  is in place, over being member of coalition  $S_2$  when  $\Pi'$  is in place, or at least,  $i$  prefers both coalition/partition pairs equally (when the preference is strict, it is denoted by  $\succ_i$ ). Note that a preference relation can be used to compare two coalitions in the same partition, or the same coalition in two different partitions. For the peer-to-peer cooperation game, we propose the following preference relation for any peer  $i \in \mathcal{N}$ :

$$(S_1, \Pi) \succeq_i (S_2, \Pi') \Leftrightarrow w_i(S_1, \Pi) \geq w_i(S_2, \Pi'), \quad (6)$$

where  $S_1 \in \Pi$ ,  $S_2 \in \Pi'$ , with  $\Pi, \Pi' \in \mathfrak{B}$ , are any two coalitions that contain peer  $i$ , i.e.,  $i \in S_1$  and  $i \in S_2$  and  $w_i$  is a preference function defined for a peer  $i \in \mathcal{N}$  as follows:

$$w_i(S, \Pi) = \begin{cases} \phi_i(S, \Pi), & \text{if } \phi_j(S, \Pi) \geq \phi_j(S \setminus \{i\}, \Pi), \\ & \forall j \in S \setminus \{i\} \& S \notin h(i) \text{ or } (|S| = 1), \\ 0, & \text{otherwise,} \end{cases} \quad (7)$$

where  $\phi_i(S, \Pi)$  is given by (3) through (5) and it represents the delay perceived by peer  $i \in S$  when partition  $\Pi$  is in place and  $h(i)$  is the history set of peer  $i$  which holds the coalitions of size larger than 1 that  $i$  was member of in the past, and had parted.

The main idea behind the function  $w_i$  is that a peer  $i$  assigns a preference equal to its achieved payoff for any coalition/partition pair  $(S, \Pi)$  as long as coalition  $S$  is either a singleton coalition (i.e.,  $i$  is acting non-cooperatively) or  $S$  is a coalition of size larger than 1, which was not previously visited by  $i$  (not in  $h(i)$ ), and where the joining of peer  $i$  to coalition  $S$  is not detrimental to any of the peers already in  $S \setminus \{i\}$ . Otherwise, the peer assigns a zero preference value to any coalition whose members' payoffs decrease due to the presence of  $i$ , since the members of such a coalition will refuse to have  $i$  join the coalition. Also, peer  $i$  assigns a zero preference to any coalition it already visited and *left* since  $i$  has no incentive to revisit such a coalition.

#### IV. PROPOSED COALITION FORMATION ALGORITHM

For forming coalitions, we propose the following rule that can be followed individually by each peer in the network:

**Definition 3: Change Rule** - For a given partition  $\Pi = \{S_1, \dots, S_M\}$  of the set of peers  $\mathcal{N}$ , a peer  $i$  can decide to *change* its coalition, i.e., leave its current coalition  $S_m$ , for some  $m \in \{1, \dots, M\}$  and join another coalition  $S_k \in \Pi \cup \{\emptyset\}$ , with  $S_k \neq S_m$ , if and only if  $(S_k \cup \{i\}, \Pi') \succ_i (S_m, \Pi)$  where  $\Pi' = \{\Pi \setminus \{S_m, S_k\}\} \cup \{S_m \setminus \{i\}, S_k \cup \{i\}\}$  is the partition resulting of the change. This change rule is represented by  $\{S_m, S_k\} \rightarrow \{S_m \setminus \{i\}, S_k \cup \{i\}\}$  and  $\Pi \rightarrow \Pi'$ .

Using the change rule, any peer can decide to leave its present coalition  $S_m \in \Pi$  and join a new coalition  $S_k \in \Pi$ , forming a new partition  $\Pi'$ , as long as the new pair  $(S_k \cup \{i\}, \Pi')$  is strictly preferred over  $(S_m, \Pi)$  through the preference relation defined by (6) and (7). That is, a peer can move to a new coalition if it can strictly improve its payoff, *without* decreasing the payoff of any member of the new coalition, i.e., given the consent of the new members as per (6). Further, each time a peer executes a change rule from its current coalition  $S_m \in \Pi$ , coalition  $S_m$  is stored in its history set  $h(i)$  (if  $|S_m| > 1$ ).

Consequently, we propose a coalition formation algorithm composed of three main stages: Peer discovery, coalition formation, and cooperative download. During the first stage, as the seed announces the availability of the resources, all interested peers attempt to download the content from the seed. Meanwhile, the peers can use a tracker or other well known peer-discovery algorithms such as in [1], [2] to learn of the presence of other peers downloading from the seed. Once peer discovery is done, the peers engage in the coalition formation stage. In this stage, each peer attempts to estimate its payoff from changing its current coalition and joining another coalition (or peer). Once a peer finds a potential change possibility (satisfying (6) and (7)), it can make a *distributed* decision to break from its current coalition and join a new coalition. In this stage, we consider that the peers make decisions in a certain random order (dictated by who requests first to cooperate). The peers take their change decisions based on an assessment, using (3) and (4), of the payoff given the current partition and not on the long term payoff. Such a strategy is a quite common approach for decision making, referred to as *myopic* strategy [8]. Following the convergence of the coalition formation process, a partition  $\Pi_f$  is in place in the network for downloading the resources from the seed which occurs in the last stage of the algorithm using the model described in Section II. A summary of the proposed algorithm is given in Algorithm 1.

The stability of the partition  $\Pi_f$  can be studied using the following stability concept:

**Definition 4:** A partition  $\Pi = \{S_1, \dots, S_M\}$  is *Nash-stable* if  $\forall i \in \mathcal{N}$  s. t.  $i \in S_m, S_m \in \Pi$ ,  $(S_m, \Pi) \succeq_i (S_k \cup \{i\}, \Pi')$  for all  $S_k \in \Pi \cup \{\emptyset\}$  with  $\Pi' = (\Pi \setminus \{S_m, S_k\}) \cup \{S_m \setminus \{i\}, S_k \cup \{i\}\}$ .

Hence, a partition  $\Pi$  is Nash-stable, if no peer has an incentive to move from its current coalition to another coalition in  $\Pi$  or to deviate and act alone. Given this notion, the convergence of the proposed algorithm during the coalition formation stage is guaranteed as follows:

**Theorem 1:** Starting from any initial network partition  $\Pi_0$ , the coalition formation stage of the proposed algorithm always converges to a final Nash-stable network partition  $\Pi_f \in \mathfrak{B}$ .

*Proof:* This is a result of the *finite* number of partitions of the set  $\mathcal{N}$  (given by the Bell number [8]) and of the definition of the change rule. ■

**Algorithm 1** Proposed peer-to-peer coalition formation algorithm.**Initial State**

The network is partitioned by  $\Pi_0 = \{S_1, \dots, S_M\}$ . At the beginning of all time, the network is non-cooperative, hence,  $\Pi_0 = \mathcal{N}$ .

**Stage 1 - Peer Discovery:**

The seed announces the availability of resources sought by all peers.

Each peer in  $\mathcal{N}$  attempts to download the content from the seed.

Using peer discovery algorithms such as trackers or others [1], [2], the peers discover the presence of other peers at the seed.

**Stage 2 - Coalition Formation:****repeat**

Each peer  $i \in \mathcal{N}$  investigates potential change operations using the preference in (6) by engaging in pairwise negotiations with existing coalitions in partition  $\Pi$ .

Once a change operation is found:

a) Peer  $i$  leaves its current coalition.

b) Peer  $i$  updates its history  $h(i)$ , if needed.

c) Peer  $i$  joins the new coalition with the consent of its members.

**until** convergence to a Nash-stable partition**Stage 3 - Cooperative Download:**

The formed coalitions perform cooperative download from the seed and their partners as per Section II.

## V. SIMULATION RESULTS AND ANALYSIS

## A. Analysis of a network with 2 peers

In the scenario of a network with 2 peers, under certain conditions on the upload rate of the seed  $\mu^s$ , the optimal social cost minimizing solution for the coalition  $S = \{1, 2\}$ , if this coalition forms, can be given by the following theorem:

**Theorem 2:** Consider two peers seeking to download data from a common seed and having, respectively, the download and upload rates  $\lambda_1$ ,  $\lambda_2$ ,  $\mu_1$ , and  $\mu_2$ . In this scenario, given  $\lambda_T = \lambda_1 + \lambda_2$ , whenever the seed's upload rate  $\mu^s > \lambda_T$  satisfies

$$\mu^s \leq \begin{cases} (\sqrt{2} + 1)\mu_1 - \mu_2 + \lambda_T & \text{if } \mu_1 \leq \mu_2 < (\sqrt{2} + 1)\mu_1 \\ (\sqrt{2} + 1)\mu_2 - \mu_1 + \lambda_T & \text{if } \mu_2 < \mu_1 \leq (\sqrt{2} + 1)\mu_2 \end{cases} \quad (8)$$

The optimal solution that minimizes the social cost of coalition

$S = \{1, 2\}$ , if  $S$  forms, is  $\mathbf{P}_S^* = \begin{bmatrix} p_{10}^* & p_{20}^* \\ p_{12}^* & p_{21}^* \end{bmatrix}$  with

$$p_{12}^* = 1 - p_{10}^* = \frac{(\sqrt{2} - 2)(\mu_1 + \mu^s - \lambda_T) + \sqrt{2}\mu_2}{2\lambda_1}, \quad (9)$$

$$p_{21}^* = 1 - p_{20}^* = \frac{(\sqrt{2} - 2)(\mu_2 + \mu^s - \lambda_T) + \sqrt{2}\mu_1}{2\lambda_2}. \quad (10)$$

*Proof:* Consider the scenario where 2 peers, with respective upload and download rates  $\lambda_1$ ,  $\lambda_2$ ,  $\mu_1$ , and  $\mu_2$  wish to download resources from a common seed having an upload rate  $\mu^s > \lambda_T$ ,  $\lambda_T = \lambda_1 + \lambda_2$ . For analyzing whether it is of their benefit to form coalition  $S = \{1, 2\}$ , the peers need to find the social cost minimizing request rate distribution matrix  $\mathbf{P}_S^*$ , as per (4). Denote by  $x_{12} = p_{12}\lambda_1$  as the amount of download requests directed by peer 1 to peer 2 and  $x_{21} = p_{21}\lambda_2$  its counterpart from peer 2 on peer 1. Thus, given the download delays  $\tau_1(\mathbf{P}_S)$  and  $\tau_2(\mathbf{P}_S)$  of peers 1 and 2, respectively, as per (3), the total delay of the coalition  $S$  is (the dependence on  $\mathbf{P}_S$  is dropped for simplicity)

$$\tau_{\text{total}}^S(\mathbf{P}_S) = \tau_1 + \tau_2 = \frac{x_{12}}{4\mu_2(\mu_2 - x_{12})} + \frac{x_{21}}{4\mu_1(\mu_1 - x_{21})} + \tau_s \quad (11)$$

with  $\tau_s = \frac{\lambda_T - x_{12} - x_{21}}{2\mu^s(\mu^s - (\lambda_T - x_{12} - x_{21}))}$  as per (1) with  $\lambda^s = (\lambda_1 - x_{12}) + (\lambda_2 - x_{21})$ .

For finding the optimal  $\mathbf{P}_S^*$  using (4), the peers need to minimize  $\tau_{\text{total}}^S(\mathbf{P}_S)$  given the constraints

$$0 \leq p_{12} < \frac{\mu_2}{\lambda_1}, \quad 0 \leq p_{21} < \frac{\mu_1}{\lambda_2}, \quad (12)$$

$$p_{12} + p_{10} = 1, \quad p_{21} + p_{20} = 1.$$

Note that since  $\mu^s > \lambda_T$  then  $p_{10}$  and  $p_{20}$  always satisfy the constraints of (4). By solving  $\frac{\partial \tau_{\text{total}}^S(\mathbf{P}_S)}{\partial x_{12}} = 0$  and manipulating the resulting quadratic equation, we obtain, using  $\mu^s > \lambda_T$ , two roots for the equation (details are omitted due to page limits)

$$x'_{12} = (\alpha + 1)x_{21} + (\alpha + 1)(\mu^s - \lambda_T) + (\alpha + 2)\mu_2, \quad (13)$$

where  $\alpha = \pm\sqrt{2}$ . By inspecting (13), one can verify that having  $\alpha = \sqrt{2}$  and given  $\mu^s > \lambda_T$  yields  $x'_{12} > \mu_2$  which is an infeasible solution since we must have  $x_{12} \leq \mu_2$ . Hence, the only possible solution is

$$x_{12} = (1 - \sqrt{2})x_{21} + (1 - \sqrt{2})(\mu^s - \lambda_T) + (2 - \sqrt{2})\mu_2 \quad (14)$$

In a symmetric manner, by setting  $\frac{\partial \tau_{\text{total}}^S(\mathbf{P}_S)}{\partial x_{21}} = 0$ , we find

$$x_{21} = (1 - \sqrt{2})x_{12} + (1 - \sqrt{2})(\mu^s - \lambda_T) + (2 - \sqrt{2})\mu_1 \quad (15)$$

and, thus, the optimal solution is the solution of the system (14) and (15) which satisfies the constraints, and is given by

$$x_{12}^* = \frac{(\sqrt{2} - 2)}{2}\mu_1 + \frac{\sqrt{2}}{2}\mu_2 + \frac{(\sqrt{2} - 2)}{2}(\mu^s - \lambda_T) \quad (16)$$

$$x_{21}^* = \frac{(\sqrt{2} - 2)}{2}\mu_2 + \frac{\sqrt{2}}{2}\mu_1 + \frac{(\sqrt{2} - 2)}{2}(\mu^s - \lambda_T) \quad (17)$$

By substituting (16) and (17) in the constraints of (12) and through algebraic manipulations while maintaining  $\mu^s > \lambda_T$ , we find that the derived solution is feasible whenever (8) is satisfied. Finally, given  $x_{12}^* = p_{12}^*\lambda_1$ ,  $x_{21}^* = p_{21}^*\lambda_2$ ,  $p_{10}^* = 1 - p_{12}^*$ , and  $p_{20}^* = 1 - p_{21}^*$ , the optimal matrix  $\mathbf{P}_S^*$  can be computed. ■

In summary, Theorem 2 provides an analytical closed form solution that the peers can use to compute their optimal policy for download request distribution, given that the seed's upload rate satisfies (8)<sup>1</sup>. Moreover, once the two peers compute the optimal divisions for the coalition  $S = \{1, 2\}$ , using (6) and (7) (as per the change rule), the peers would agree to join into a single coalition  $S$  if at least one peer is better off in the cooperative case without decreasing the payoff of the other peer, i.e.,  $\tau_1(\mathbf{P}_S^*) \geq \tau_{\text{nc}}$  with  $\tau_2(\mathbf{P}_S^*) > \tau_{\text{nc}}$  or  $\tau_1(\mathbf{P}_S^*) > \tau_{\text{nc}}$  with  $\tau_2(\mathbf{P}_S^*) \geq \tau_{\text{nc}}$  where  $\tau_{\text{nc}}$  is the non-cooperative delay as per (1).

As seen by (3), (9), and (10), for a given seed load  $\mu^s > \lambda_1 + \lambda_2$ , the decision of the two peers to join inherently depends on the characteristics of the peers, i.e., their download and upload rates. For instance, from Theorem 2, it is interesting to note that the optimal divisions  $p_{12}^*$  and  $p_{21}^*$  are linear functions of the upload rates  $\mu_1$ ,  $\mu_2$ , and  $\mu^s$ . Further, given a fixed upload rate for a peer, e.g.,  $\mu_1$ , as the upload rate  $\mu_2$  of the other peer increases, the social cost minimizing solution dictates that peer 1 leeches less and less on peer 2. In order to emphasize this aspect and show how the upload rate of the peers affects their cooperative policies (9) and (10), we consider the following example.

**Example 1:** Consider 2 peers with, respectively, the download and upload rates  $\lambda_1 = 1.4$  Mbps,  $\lambda_2 = 1$  Mbps,  $\mu_1 = 512$  kbps, and  $\mu_2 \in [256, 850]$  kbps. We consider the case when a server, acting as a seed, with an upload rate  $\mu^s = 2.5$  Mbps announces the availability of resources that peers 1 and 2 seek to download. These parameters are chosen such that, at all  $\mu_2$ , (8) is verified and both peers find it always beneficial to form a single coalition as per the change rule and (6). For this scenario, we are interested in the variation of the optimal policies of the peers, as per (9) and (10), when coalition  $S = \{1, 2\}$  forms and

<sup>1</sup>Alternatively, in the case when (8) is not satisfied, the peers can always find their optimal download distributions using classical optimization techniques [9].

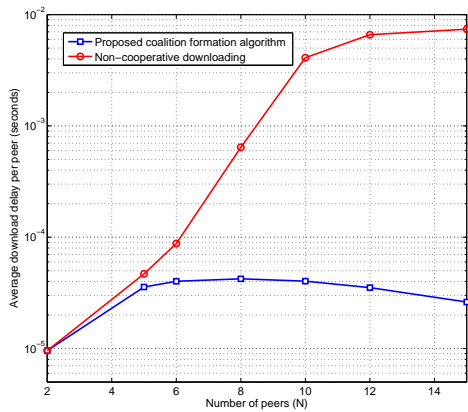


Fig. 1. Average download delay per peer achieved by the proposed algorithm and the non-cooperative case as the number of peers  $N$  increases.

as  $\mu_2$  changes. By inspecting (9), we can see that, as the upload capability of peer 2 increases, peer 1 tends to put more download requests on peer 2, while peer 2 decreases its download requests on peer 1. For instance, while at  $\mu_2 = 256$  kbps, peer 1 does not download any data from peer 2 as  $p_{12}^* \approx 0$  as per (9) (but still benefits from cooperation due to the reduced load on the server from the cooperation of peer 2), at  $\mu_2 = 850$  kbps, peer 1 download up to  $p_{12}^* = 30\%$  of its data from peer 2. In contrast, the fraction of download of peer 2 from peer 1 decreases from around  $p_{21}^* = 25.8\%$  at  $\mu_2 = 256$  kbps down to  $p_{21}^* = 8.4\%$  at  $\mu_2 = 850$  kbps. Hence, as one of the peers leeches more on the other, the optimal policy of the other peer dictates to download more from the server (which is less congested due to the cooperative behavior of the other peer) and less from the other peer. Finally, by solving the system of equalities constituted by (9) and (10), we can find that, at  $\mu_2 = 577.5$  kbps, the optimal policy of both peers is to equally download from each other, i.e.,  $p_{12}^* = p_{21}^* = 16.3\%$ .

### B. Simulation results for a network with $N \geq 2$ peers

For simulations, we consider a server as a seed with a robust connection having, unless stated otherwise, an upload rate of  $\mu^s = 15$  Mbps. At the time the server announces the availability of resources for download,  $N \geq 2$  peers in the set  $\mathcal{N}$  with download rates  $\lambda_i, \forall i \in \mathcal{N}$ , attempt to concurrently download the content from the server. For all simulations, the download rates  $\lambda_i, \forall i \in \mathcal{N}$  are randomly generated from a uniform distribution over  $[256, 3000]$  kbps while maintaining the queueing stability condition  $\mu^s > \sum_{i \in \mathcal{N}} \lambda_i$ . Further, all the peers' connections are considered asymmetric with upload rates uniformly distributed in the interval  $\mu_i \in [256, \min(\lambda_i, 1024)]$  kbps,  $\forall i \in \mathcal{N}$ . All results are *averaged*, through a large number of runs, over all possible random choices of the peers' download and upload rates.

In Fig. 1, we show the average download delay per peer as the number of peers downloading from the server  $N$  increases. Fig. 1 shows that, as  $N$  increases, the average delay for the non-cooperative scheme increases due to the increased load on the server. In contrast, for the proposed algorithm, although the average delay starts by increasing slowly, at  $N = 10$  peers, this average delay starts to slightly decrease with  $N$  since the benefit from cooperation grows due to: (i)- The presence of more peers to cooperate with and (ii)- The growing need for cooperation with the increasing server load. Fig. 1 demonstrates that the proposed algorithm presents a significant performance advantage, in terms of average download delay per peer, increasing with the numbers of peers  $N$  (due to the presence of more

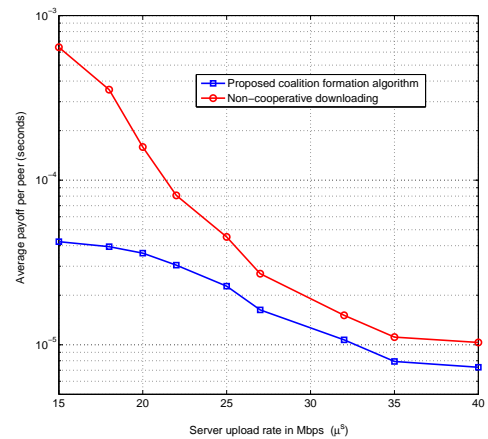


Fig. 2. Average download delay per peer for a scenario with  $N = 10$  peers as the server upload rate  $\mu^s$  varies.

cooperation possibilities) and reaching up to 99.6% (two orders of magnitude) improvement relative to the non-cooperative case at  $N = 15$  peers.

In Fig. 2, we show the average download delay per peer for a scenario with  $N = 10$  peers as the upload rate of the server  $\mu^s$ , increases. In this figure, we can see that, for both the proposed scheme and the non-cooperative approach, as  $\mu^s$  increases, the average download delay decreases, as the server becomes less congested. Further, Fig. 2 shows that the increase of  $\mu^s$  decreases the performance advantage of the proposed algorithm as the peers would have less incentive to cooperate due to being somehow satisfied with their non-cooperative download delays (which becomes smaller as  $\mu^s$  grows). Nonetheless, Fig. 2 demonstrates that, at all  $\mu^s$ , the proposed coalition formation algorithm yields a performance advantage of at least 29.3% (at  $\mu^s = 40$  Mbps) relative to the non-cooperative case.

## VI. CONCLUSIONS

In this paper, we studied the problem of cooperation among a number of peers seeking to download a content of interest from a common seed at the time this seed announces the availability of the resources. We modeled the problem as a coalition formation game in partition form and we proposed an algorithm for forming coalitions. By engaging in the proposed algorithm, the peers can take individual decisions to join or break from a coalition while minimizing their average download delay. We demonstrated that the coalition formation process allows the peers to self-organize into a Nash-stable partition. For the 2-peer case, we derived analytical results for the optimal download request divisions. Through simulations, we showed that the proposed algorithm yields significant performance gains.

## REFERENCES

- [1] BitTorrent. [Available Online], [bitconjurer.org/BitTorrent/](http://bitconjurer.org/BitTorrent/), 2003.
- [2] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," *IEEE Communications Survey and Tutorials*, vol. 7, pp. 72–93, Mar. 2005.
- [3] D. Qiu and R. Srikant, "Modeling and performance analysis of BitTorrent-like peer-to-peer networks," in *Proc. ACM SIGCOMM*, Portland, OR, USA, Aug. 2004, pp. 367–378.
- [4] H. Park and M. van der Schaar, "A framework for foresighted resource reciprocation in P2P networks," *IEEE Trans. on Multimedia*, vol. 11, no. 1, pp. 101–116, Jan. 2009.
- [5] Y. Lien, "Performance issues of P2P file sharing over asymmetric and wireless networks," in *Proc. of the IEEE Int. Conf. on Distributed Computing Systems*, Columbus, OH, USA, Jun. 2005, pp. 850–855.
- [6] D. Bertsekas and R. Gallager, *Data Networks*. New Jersey, USA: Prentice Hall, Mar. 1992.
- [7] S. Shakkottai and R. Srikant, "Network optimization and control," *Foundations and Trends in Networking*, vol. 2, no. 3, pp. 271–379, Jan. 2007.
- [8] W. Saad, Z. Han, M. Debbah, A. Hjørungnes, and T. Başar, "Coalitional game theory for communication networks," *IEEE Signal Processing Mag.*, vol. 26, no. 5, pp. 77–97, Sep. 2009.
- [9] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, USA: Cambridge University Press, Sep. 2004.