Proportional fairness in heterogeneous peer-to-peer networks through reciprocity and Gibbs sampling

Martín Zubeldía, Andrés Ferragut and Fernando Paganini
Universidad ORT Uruguay

***Abstract*— This paper studies peer-to-peer networks with the objective of imposing a proportionally fair allocation of peer upload capacity. We begin with a tutorial review on the feasibility of achieving these allocations with idealized assumptions on connectivity and rate control, as well as a distributed algorithm based on peer reciprocity that can achieve it. To impose some of the constraints of real networks (limited number of connections, with bandwidth imposed by lower layers) we introduce an energy function that measures the deviations from ideal reciprocity, and analyze methods to minimize this energy in a decentralized way. To avoid combinatoric difficulties, as well as to enable new peer exploration, we use a Gibbs sampler approach, in which a Markov chain is designed with stationary distribution determined by our energy function. This proposal is implemented and tested in simulation, and results are compared with other existing and proposed P2P exchange systems.**

I. Introduction

In peer-to-peer (P2P) file sharing networks, scalability of the service capacity is obtained by requiring each downloading client to become a server to others, contributing upload bandwidth. Consider a scenario where a set of peers upload at rates $\mu_i, i = 1, \dots N$; assume there are no other bottlenecks in the network, and enough diversity of pieces so that the total bandwidth $\sum_i \mu_i$ can be used for download; how should it be distributed among the same peers, now seen as clients?

This question has been addressed by many researchers, as will be reviewed below; often, the discussion is combined with efforts to characterize the behavior of prevailing P2P protocols such as BitTorrent [5]. In this paper we follow the route of selecting a desirable objective, studying first its feasibility in ideal terms, and progressively imposing the design constraints of practical systems. In this process we will provide a tutorial review of earlier literature, add some new results, and develop a new proposal, which is then analyzed mathematically and tested in simulation.

In general, there could be a tradeoff between *performance* and *fairness* in the allocation rule, as studied in [7]; however, in the scenario (dominant in practice) of only uplink bottlenecks the second issue prevails. In this regard, a natural, simple answer to our question is: you should get as much as you give [11]. This proportionally fair [17] allocation provides direct, transparent incentives for peers to contribute [9]; additional interpretations are given in Section II. Its feasibility can be studied by writing the mutual peer exchange bandwidths in matrix form [16]: the question reduces to a problem of matrix row and column renormalization, studied classically by Sinkhorn [13]–[15]. Indeed, the natural iteration of renormalizing rows and columns leads to a *reciprocity* algorithm that can achieve the desired allocation, whenever feasible.

This ideal reciprocity scheme is not easily taken to practice. In Section III we consider two important implementation restrictions: (i) for overhead reasons, peers must maintain simultaneous connections with only a small amount of peers; and, (ii) due to the underlying TCP protocol, these connections will receive an equal share of the peer's upload bandwidth. To study these limitations we introduce an *energy function* which is zero under ideal reciprocity, and which a practical scheme should try to reduce. Although optimizing energy under constraints (i)-(ii) has combinatoric complexity, we identify cases where zero energy is indeed achievable; more generally, we characterize the algorithm in which each peer tries to myopically reduce its portion of the energy: a tit-for-tat structure similar to BitTorrent's comes out naturally from this procedure.

The final step in the implementation road is to introduce some randomness in peer selection, which avoids traps in a deterministic myopic algorithm, and also explores the set of peers which will in practice vary in time. For this task we turn in Section IV to a Gibbs' sampler [4], designing a Markov process guided by a potential defined in terms of our energy function. In this regard, we note that recent papers [12], [18] have introduced this technique in the study of P2P systems from a network utility maximization perspective. As we will explain, there are differences between the two proposals, reflected in the potential functions used.

The resulting neighbor selection algorithm was tested in simulation and compared to other existing protocols: standard BitTorrent implementations, the PropShare algorithm of [9] which also aims for proportional reciprocity, as well as the proposal in [12], performing well against the alternatives in terms of reciprocity and fairness. Results are reported in Section V and conclusions are given in Section VI.

II. Proportional fairness and reciprocity

In this section we analyze the allocation objective outlined above (receive as much as you get) from the point of view of its feasibility, as well as the search for a decentralized peer reciprocity scheme that can reach this allocation.

We begin by defining some notation. A set of N peers shares information through a connectivity graph G: two peers are neighbors in this graph if they can exchange information.

E-mail:zubeldia@ort.edu.uy.

This work was supported in part by ANII-Uruguay under grant FCE_2_2011_1_7052 and under scholarship POS_NAC_2012_1_9088.

Let $A = (a_{ij})$ be the adjacency matrix of the graph, assumed symmetric. Note that in BitTorrent parlance, we are only modeling the behavior of *leechers*, who are both uploading and downloading content. We note that $a_{ii} = 0\ \forall i$.

We model the bandwidth sharing by a matrix $Z \in \mathbb{R}_+^{N\times N}$ in which the entry z_{ij} corresponds to the throughput of the connection from peer i to peer j. The matrix Z has the following properties:

$$z_{ij} = 0 \text{ if } a_{ij} = 0, \qquad \sum_j z_{ij} = \mu_i\ \forall i, \tag{1}$$

where we recall μ_i is the total upload rate of peer i.

On the other hand, the received bandwidth per peer is obtained through the column sums

$$r_j(Z) = \sum_i z_{ij} \quad \forall j.$$

The question that arises is how to allocate the total upload capacity $\sum_i \mu_i$ among all the peers. This problem is different from most graph-based resource allocation problems, as the bottlenecks are in the nodes instead of in the edges.

A. Proportional allocation

We consider as target allocation the situation where each peer receives the same bandwidth that it gives to the network, that is $r_j = \mu_j\ \forall j$. This property was called *global proportional fairness* in [17].

We argue that this rule provides the correct incentives to contribute to the network. Clearly, a minimal fairness incentive is the weaker statement that the rates r_j increase with the μ_j, i.e. no one receives less when contributing more. However since by construction $\sum_j r_j = \sum_j \mu_j$, the easiest way to achieve monotonicity is to set $r_j = \mu_j$. An alternative, economic viewpoint is to say that all peers are trading a single commodity, with a single price, hence the market equilibrium requires all individual trades to balance out. For further discussion of the incentives in proportional allocation, including resistance to attacks, we refer to [9]. See also [1], [10] for related game-theoretic studies.

A first question is whether a matrix Z exists satisfying $r_j(Z) = \mu_j$, i.e. with prescribed row and column sums; we will call this a *feasible* allocation. A characterization of this feasibility was given in [2]:

Proposition 1: Given an adjacency matrix A and a vector of capacities μ, the following are equivalent:

(i) there exists a matrix Z with column a row sums μ and with at least the same zero structure as A;
(ii) if A has a zero minor defined by subsets of rows R and columns C, then

$$\sum_{i\in R^c} \mu_i \geq \sum_{j\in C} \mu_j.$$

As a special case, if one only imposes a zero diagonal structure (that is, the network graph G is complete), feasibility will hold provided no single μ_i is greater than the sum of the rest. This is a mild restriction if the peer population is large.

We note also that a feasible allocation, if it exists, solves the optimization problem

$$\max \sum_j \mu_j \log(r_j(Z)), \text{with } Z \text{ subject to (1)};$$

in the language of utility maximization this is a weighted proportional fairness criterion. To see the above, note that the above maximization is equivalent to minimizing the Kullback-Leibler divergence (see e.g. [3])

$$D(\mu||r) = \sum_j \mu_j \log\left(\frac{\mu_j}{r_j}\right),$$

which is always non-negative (recall $\sum_j r_j = \sum_j \mu_j$) and zero when $r = \mu$.

Remark 1: This also suggests that when $r = \mu$ is infeasible, the above optimization may be a desirable objective.

B. Proportional reciprocity and the Sinkhorn iteration

The next important issue is whether the above allocation admits a decentralized implementation, i.e. a set of mutual exchange rules peers can follow to achieve it, without the intervention of a central authority. In this section we review a proportional reciprocity scheme that can achieve the proportionally fair allocation, assuming a fine control of mutual rates.

Let k be a discrete-time index that represents an exchange slot, and let $z_{ij}^{(k)}$ denote the bandwidth devoted by peer i to peer j in the k-th slot. That is, for each k we have an allocation matrix $Z^{(k)}$ for this network. Based on received rates, peers must select their allocation for the following slot; a natural rule considered in [9], [16], [17] is *proportional reciprocity*: give to others in proportion to what is received from them. Mathematically

$$z_{ij}^{(k+1)} = \mu_i . \frac{z_{ji}^{(k)}}{r_i^{(k)}},$$

or $Z^{(k+1)} = \mathrm{diag}(\mu_i / r_i^{(k)})[Z^{(k)}]^T$. This means to transpose the matrix and renormalize rows to have sum μ. Equivalently, one could first renormalize the columns to have sum μ^T, and then take transpose. In this sense, except for the transpose operation, this iteration amounts to an iterative row and column renormalization of a non-negative matrix, a topic that was studied classically by Sinkhorn [13]–[15] who established conditions for convergence. This connection, found in [16], makes it possible to obtain several results for the proportional response iteration from the extensive literature that studied the Sinkhorn renormalization (see [2] and references therein). The conditions for the convergence of the Sinkhorn algorithm are presented in the following proposition.

Proposition 2: If the scaling problem with adjacency matrix A and capacities μ is *feasible*, then for any initial matrix Y (with the zero structure of A) the Sinkhorn iteration converges to a matrix Z which is a proportional allocation.

Proportional allocations are not, in general, unique, and the limit point Z will depend on the initial condition. The following result gives a precise characterization.

Proposition 3: Under the conditions of Proposition 2, the limit allocation for initial condition Y is the solution to the following convex optimization problem:

$$\begin{aligned} \min\ D(Z||Y) := & \sum_{i=1}^{N}\sum_{j=1}^{N} z_{ij}\log\left(\frac{z_{ij}}{y_{ij}}\right) \qquad (2)\\ s.t.\quad & \sum_{j=1}^{N} z_{ij} = \mu_i \ \ \forall i = 1,\dots,N \\ & \sum_{i=1}^{N} z_{ij} = \mu_j \ \ \forall j = 1,\dots,N \\ & z_{ij} \geq 0 \quad \forall i,j. \end{aligned}$$

Interestingly, the limit point is the closest to the initial condition in K-L divergence (for the matrices, componentwise, both matrices having the same overall sum) within the feasible set.

While the Sinkhorn iteration may converge under the feasibility condition, we cannot say the same for the proportional response. The transposing of the matrices could make the sequence oscillate between a matrix and its transpose. Fortunately, if the initial matrix is symmetric, then the Sinkhorn iteration will converge to a symmetric proportional allocation and so will the proportional response dynamics.

Corollary 4: If the initial matrix Y is symmetric, then the limit matrix of the Sinkhorn iteration is symmetric.

Proof: Let Z_{opt} be the optimal matrix in (2). Since Y symmetric, the cost is a symmetric function of Z and so are the constraints, therefore Z_{opt}^T is also optimal. But the objective function is strictly convex and thus it has a unique minimum, therefore $Z_{opt} = Z_{opt}^T$. ■

It has further been shown in [16] that the even and odd subsequences of the Sinkhorn algorithm always converge, regardless of feasibility, and thus the even and odd iterations of the proportional reciprocity algorithm also converge to some limits. When the problem is not feasible, the Sinkhorn iteration oscillates between two matrices in the limit. To the best of our knowledge, it is still an open question whether these matrices are one the transpose of the other. In that case, the proportional response dynamics would converge even if the Sinkhorn iteration does not.

III. Implementation restrictions: discrete connections

The proportional response is a decentralized algorithm that achieves the desired allocation in a P2P network. However, problems arise if we wish to implement such algorithm in practice. First of all, it needs a constant connection with each neighbor peer in the network; this is impractical as there would be too many active connections, which leads to more overhead than is desired. Secondly, each connection would have to be fine-tuned to a desired rate. This is difficult to achieve, specially if you are planning on using TCP as underlying protocol. Finally, this is a completely deterministic algorithm and as such lacks the necessary randomness to explore the different peering options as the network evolves.

Before moving on to incorporate such restrictions in the analysis, we briefly review how things are handled in BitTorrent [5], the most popular P2P protocol. BitTorrent peers open a maximum amount (usually four) of connections to other peers. Using TCP connections, under normal circumstances (bottleneck in the upload) this leads to a uniform split of its bandwidth between them. The main source of (uncontrolled) differences between TCP rates would be round-trip-times; we will ignore this issue in what follows.

The resource allocation results from the neighbor selection algorithm, which has essentially two parts:

- The tit-for-tat part: in which each peer, every 10s, decides to connect only with the three peers that gave it the most in the last 20s.
- The optimistic unchoke: in which each peer, every 30s, opens a connection to a random peer for 30s.

The result is that every peer is at any time only connected to 4 peers at most, 3 of them that are chosen based on a ranking of the received bandwidths and the other one at random.

This algorithm, although practical and easy to implement, yields an allocation which is not proportional in most cases. While the tit-for-tat portion is a form of reciprocity, it falls short of the desired proportionality: as shown in [9], it amounts to a bandwidth auction and as a result, a peer has only the incentive to contribute with the minimum amount of bandwidth that will win the auction, and nothing more. Furthermore, it does not always give the incentive for peers to contribute to the network, since the optimistic unchoke portion, meant to ensure randomness, gives each peer a lower bound on received bandwidth from other peers (at least one fourth), regardless of its contribution. This breaks the reciprocity incentives and makes room for free riders.

There is thus room left for exploring alternatives to the BitTorrent neighbor selection, that will more closely reflect our design objective of proportional allocation, within the practical constraints that have been identified. In this Section we will address two of these constraints, which stem from the discrete nature of connections and reliance of transport protocols that impose bandwidth sharing:

1) Each peer can only open a maximum amount of N_0 connections.
2) The upload capacity of each peer is equally distributed between all outbound connections.

We postpone to the following section the issue of incorporating randomness in peer selection. Throughout this section, N will denote the number of peers, with their upload capacities in decreasing order: $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_N$. Again, it is assumed there are no other bottlenecks in the network.

A. Energy driven allocations

As a means to study the impact of the above discrete constraints on the desired reciprocity, we will introduce an *energy function* $\mathcal{E}'(Z)$, sum of squares of the peerwise discrepancies in exchange rates, as follows:

$$\mathcal{E}'(Z) = \frac{1}{2}\sum_{i,j}(z_{ij} - z_{ji})^2.$$

This function is defined over the set of allocation matrices Z which we recall always satisfy the restrictions in (1), for a given vector of upload capacities μ and a given adjacency matrix A, assumed symmetric.

Proposition 5: If the row and column scaling problem with adjacency matrix A and capacities μ is feasible, then the allocations of minimal energy $\mathcal{E}'(Z) = 0$ are precisely the symmetric proportional allocations

$$M^* = \left\{ Z \text{ satisfies (1)}, Z = Z^T \right\} \neq \emptyset.$$

The above follows from the theory reviewed in Section II. The set M^* is convex and is the intersection of the limit sets of the Sinkhorn algorithm and the proportional response dynamics, which make this a proper energy for our purposes.

We now begin to incorporate the discrete restrictions imposed by the number N_0 of peer connections, and the equal bandwidth between them. At this point it is convenient to factor out the peer bandwidths and introduce a matrix X with coefficients in $\{0, \frac{1}{N_0}\}$ that stores the neighboring configurations in terms of the fractions x_{ij} of its own bandwidth that peer i allocates to each peer j. From it, the rate allocation can be obtained as

$$Z = \operatorname{diag}(\mu_i)\, X.$$

Based on this, we can redefine the energy as a function of the neighboring configurations X

$$\mathcal{E}(X) = \frac{1}{2} \sum_{i,j} (\mu_i x_{ij} - \mu_j x_{ji})^2. \tag{3}$$

We would like to minimize the energy $\mathcal{E}(X)$ with the incorporated restrictions. The minimization now is over the subset of stochastic matrices

$$\Lambda^S = \left\{ X \in \left\{0, \tfrac{1}{N_0}\right\}^{N^2} : x_{ij} = 0 \text{ if } a_{ij} = 0; \sum_{j \in S} x_{ij} = 1 \right\}$$

In general, there is no explicit solution for this discrete optimization problem, but in certain cases we can find properties of the solution. One such case is where there are repeated values in the sequence of upload bandwidths $\{\mu_i\}$, of enough multiplicity with respect to the connectivity parameter N_0. We state the following result:

Proposition 6: Suppose that N_0 is even. Divide the set of peers into K groups with the same upload bandwidth $\mu^{(k)}$ for each member of group k. If every group has $N_k > N_0$ peers, there exists at least one configuration X^* such that $\mathcal{E}(X^*) = 0$, resulting in the proportional allocation.

Proof: As we have groups of peers with the same bandwidth, we could hope to form independent sets of peers with the same bandwidth connected to each other, but disconnected from the rest, thus obtaining a configuration X^* with $\mathcal{E}(X^*) = 0$. Equivalently, for each group of N_k peers we have to find a N_0-regular graph (undirected, where every node has N_0 neighbors). Fortunately, the existence of such graphs is a known result in graph theory when N_0 is even [6] (for instance, a solution is a so-called Cayley graph). As a result, every group of N_0-regular graphs would make the energy equal to 0 and thus yield a proportional allocation. ■

Remark 2: A N_0-regular graph is fundamentally different to the formation of *cliques* (complete subgraphs) which has been shown to be a property of the BitTorrent tit-for-tat mechanism [7]. A N_0-order clique has by definition $N_0 + 1$ nodes; so unless the cardinality of the repeated bandwidths happens to coincide with this value, the result will be different. In fact, an algorithm that forces cliques of fixed size can lead to severe loss in proportional reciprocity, as portrayed in the following example.

Example 1: Suppose that $N_0 = 4$ and $N = 15$, where seven peers have $\mu_i = 10$ and the other eight have $\mu_i = 1$. The method of Proposition 6 forms two regular graphs and achieves proportional reciprocity. If instead we form 3 cliques of size $N_0 + 1 = 5$, only two of these can involve homogeneous peers and deliver proportional reciprocity. The third clique will have two fast peers ($\mu_i = 10$) and three slow peers ($\mu_i = 1$), resulting in an allocation of $r = 3.25$ for the fast peers, and $r = 5.5$ for the slow ones. Not only is proportionality broken, but the fast peers are being penalized!

One might think that having exact repetition of the upload bandwidths is a very special case. However, if peers can be grouped in classes with *approximately* equal bandwidth, we can bound the minimum energy as follows.

Proposition 7: Suppose that N_0 is even. Divide the set of peers into K groups, where the bandwidths $\{\mu_i\}$ for peers in each group occupy an interval of length δ. If every group has $N_k > N_0$ peers, there exists at least one configuration X^* such that $\mathcal{E}(X^*) \leq \delta^2 \frac{N}{2N_0}$.

Proof: Consider the same X^* constructed in Proposition 6. Write the total energy as $\mathcal{E}(X^*) = \sum_k \mathcal{E}_k(X^*)$, adding the energy contributions of each disconnected group. For group k we have $N_k N_0$ mutual connections, each with energy

$$\frac{1}{2}(\mu_i x_{ij} - \mu_j x_{ji})^2 \leq \frac{\delta^2}{2N_0^2}.$$

Therefore $\mathcal{E}_k(X^*) \leq \delta^2 \frac{N_k}{2N_0}$ and the result follows from $\sum_k N_k = N$. ■

Thus suggests that grouping peers in subsets of similar bandwidth, of any size greater than N_0, is a good strategy to approximate the goal of proportional reciprocity. The size of the classes will be a function of the existing set of μ_i's; the flexibility of going beyond cliques of size $N_0 + 1$ can lead to significant improvements.

B. Decentralized energy minimization and tit-for-tat

The question to ask at this point is: can the energy by minimized by a *decentralized* algorithm? Given the combinatoric nature of the problem we do not expect the global optimum to be computable, but a reasonable heuristic is to have each peer i choose its outgoing connections seeking to myopically reduce its own portion of the energy,

$$\mathcal{E}_i(X) := \sum_j (\mu_i x_{ij} - \mu_j x_{ji})^2.$$

In this minimization we assume given the rates x_{ji} received by peer i, and we introduce the notation $J^{in} = \{j : x_{ji} \neq 0\}$ for the set of peers from which peer i is currently receiving data. Let N^{in} be the cardinality of this set, and note that there are no a priori constraints on it, in principle $0 \leq N^{in} \leq N-1$.

Since peer i will divide its bandwidth uniformly among its N_0 outgoing connections, the myopic optimization is just to choose the set $J^{out} = \{j : x_{ij} \neq 0\}$, of cardinality N_0, to minimize the energy portion $\mathcal{E}_i(X)$. The following proposition characterizes the optimal configuration.

Proposition 8: Given a set J^{in} of peers uploading to i, a configuration X^* minimizes the local energy $\mathcal{E}_i(X)$ if and only if it solves

$$\max_{J^{out}} \sum_{J^{in} \cap J^{out}} \mu_j. \tag{4}$$

Proof: For convenience we will denote by $\tilde{\mu}_j := \frac{\mu_j}{N_0}$, the fraction of bandwidth allocated in a single connection from peer j. The local energy of a given configuration X can then be expressed as follows:

$$\mathcal{E}_i(X) = \sum_{j \in J^{in} \cap J^{out}} (\tilde{\mu}_i - \tilde{\mu}_j)^2 + \sum_{j \in J^{in} \setminus J^{out}} \tilde{\mu}_j^2 + \sum_{j \in J^{out} \setminus J^{in}} \tilde{\mu}_i^2.$$

Expanding the square $(\tilde{\mu}_i - \tilde{\mu}_j)^2 = \tilde{\mu}_i^2 + \tilde{\mu}_j^2 - 2\tilde{\mu}_i\tilde{\mu}_j$ and rearranging terms leads to the equivalent expression

$$\mathcal{E}_i(X) = \sum_{j \in J^{in}} \tilde{\mu}_j^2 + \sum_{j \in J^{out}} \tilde{\mu}_i^2 - 2 \sum_{j \in J^{in} \cap J^{out}} \tilde{\mu}_i \tilde{\mu}_j.$$

The first term above is given, and the second is fixed at $N_0\tilde{\mu}_i^2$ for all allowable configurations, so only the third term can be minimized by choice of J^{out}; noting that $\tilde{\mu}_i$ is fixed, and $\mu_j = N_0\tilde{\mu}_j$, we arrive at the equivalent maximization (4). ■

To interpret the max-weight type condition (4), we distinguish two cases:

(i) $N^{in} \leq N_0$. In this case it is clearly optimal in (4) to cover the entire set J^{in} with J^{out}, assigning any extra elements arbitrarily.

(i) $N^{in} > N_0$. In this case only a portion of the μ_j can be included. The maximum weight is achieved by assigning J^{out} to the largest N_0 values of $\{\mu_j, j \in J^{in}\}$.

So we see that the local reciprocity energy is minimized by picking N_0 peers that are currently giving the most bandwidth to peer i, and assigning any extra slots arbitrarily. Interestingly, this corresponds exactly to the tit-for-tat part of the BitTorrent algorithm. Therefore, the myopic optimization of our energy cost is consistent with this widespread reciprocity mechanism.

What happens if we iterate on the above deterministic algorithm, each peer successively updating its configuration based on the tit-for-tat like reciprocity scheme? In general, it is difficult to characterize the behavior of such dynamics over a discrete set of configurations. The trajectory will depend on initial conditions, and there is no reason to expect the global energy-minimizing configuration will be found.

For example, the initial file-exchange may break the graph into components, leaving some peers disconnected from their optimal neighbors; these will never be discovered by the above deterministic reciprocity. This suggests that a certain amount of random exploration is required. BitTorrent addresses this issue through the optimistic unchoke portion; however this egalitarian file-sharing implies an important deviation from proportionality. An alternative is studied in the following section.

IV. Incorporating randomness through the Gibbs sampler

The fact that we have established a configuration *energy* that we are attempting to minimize, suggests introducing randomness by means of a Markov process on the set of configurations, designed so that its invariant distribution can be explicitly computed and concentrated on states of low energy. This approach is often termed a Gibbs sampler [4] and the corresponding law a Gibbs distribution. Another commonly used name is "Glauber dynamics".

We remark at this point that in recent work by [12], [18], it was proposed to use this kind of approach for a P2P network utility maximization problem, and it was argued that this "reverse engineered" BitTorrent. In this regard, we make the following remarks:

- The energy function used in the Gibbs approach of [12], [18] is defined in terms of a network utility, aimed more at performance than at fairness. This would have impact in a situation where the rate of upload of peer i is not equivalent for all peers j, due to other network bottlenecks.
- The dynamics proposed in these references implies *choking* one of the current peers and replacing by a new one; the peer most likely to be choked is the one with lowest current rate *to* it in the *upload* sense. Such a rule is in fact consistent with the algorithm for *seeders* in the BitTorrent protocol (peers who already own the file). It is different, however, to a reciprocity scheme based on *download* rates received *from* other peers, as in the tit-for-tat mechanism used by *leechers*. The latter is the focus of our work, and so our Gibbs proposal will be complementary to these references.

We overview the main concepts of Gibbs measures as applied to the problem at hand, relying extensively on the reference [4]. A Gibbs distribution on a finite space of configurations X is a probability measure

$$\pi^T(X) = \frac{1}{Z_T} e^{-\frac{\mathcal{E}(X)}{T}},$$

where $\mathcal{E}(X)$ is a potential energy function, and the real parameter is T called the "temperature". Here Z_T is a suitable normalization constant. From the chosen form it is intuitively clear that as the temperature becomes lower, the distribution becomes concentrated on the minima of $\mathcal{E}$. The following is a known result in this regard.

Proposition 9: Let $\{X_1^*, \ldots, X_K^*\}$ be the set of configurations that minimize the energy $\mathcal{E}(X)$, then as $T \to$

0^+ the distribution π^T converges to $\sum_{i=1}^K \frac{1}{K}\delta_{X_i^*}$, uniform distribution on the optimal set.

The above outlines an approximate method for a general optimization problem over a discrete set X: construct a Markov chain on X in such a way that the stationary distribution turns out to be π_T, for small T.

We are interested, however, in a special case of the above procedure with a graph structure, and where the Markov chain results from neighbor interactions. In our case this graph will represent peer interactions.

Consider a finite set of *sites* S (the peers) and a graph G with nodes in S and edges describing the allowable interactions. In our case, the graph G has the adjacency matrix A as described previously. The configuration state X is obtained by assigning to each site $i \in S$ a row vector $X_i \in \{0, \frac{1}{N_0}\}^N$, with unit sum and $x_{ij} = 0$ whenever $a_{ij} = 0$ (in other words, a vector of unchoked peers). The overall configuration space coincides with Λ^S given before.

Transitions in configuration space must be based on decentralized information to each site. In the Gibbs' theory, this requires the energy to be defined in a special way. First, assign a *potential* to subgraphs of G, which must be zero except for cliques. The sum of all these potentials defines the energy. In our case, we will assign a potential V_C only to the two-node cliques $C = \{i, j\}$, by

$$V_C(x) = (\mu_i x_{ij} - \mu_j x_{ji})^2$$

The resulting energy is the sum over all such cliques,

$$\mathcal{E}(x) = \sum_{C \subset S} V_C(x) = \frac{1}{2} \sum_{i,j \in S} (\mu_i x_{ij} - \mu_j x_{ji})^2,$$

which coincides with our definition in (3). The Gibbs probability measure based on this energy is

$$\pi^T(X) = \frac{\exp\left(-\frac{1}{2T} \sum_{i,j \in S} (\mu_i x_{ij} - \mu_j x_{ji})^2\right)}{\sum_{X' \in \Lambda^S} \exp\left(-\frac{1}{2T} \sum_{i,j \in S} (\mu_i x'_{ij} - \mu_j x'_{ji})^2\right)}.$$

A. Random sweep Gibbs sampler

We now define a continuous time Markov chain which has stationary distribution π^T and only involves neighbor interactions. The only transitions that are admissible are between configurations X and X' that only differ in one row, that is, in the connections of one peer. Given X, denote by $\Lambda_i^S(X) = \{X'' \in \Lambda^S : x''_{kj} = x_{kj}, \forall k \neq i, \forall j\}$, that is, all the possible configurations that can be reached from X changing only row i. For any $X' \in \Lambda_i^S(X)$, define the transition rate

$$q^T_{X,X'} = \tau \cdot p^T_{X,X'}, \text{ where} \tag{5}$$

$$p^T_{X,X'} = \frac{\exp\left(-\frac{1}{T} \sum_{j \in S} (\mu_i x'_{ij} - \mu_j x_{ji})^2\right)}{\sum_{X'' \in \Lambda_i^S(X)} \exp\left(-\frac{1}{T} \sum_{j \in S} (\mu_i x''_{ij} - \mu_j x_{ji})^2\right)},$$

and $\tau > 0$ is a parameter.

The main property of the chosen transition rates is that

$$\pi^T(X) q^T_{X,X'} = \pi^T(X') q^T_{X',X},$$

where we note that $\Lambda_i^S(X') = \Lambda_i^S(X)$ for every $X' \in \Lambda_i^S(X)$. The above *detailed balance equations* imply that the Markov chain 5 is reversible [8] and has invariant distribution π^T as required.

Additionally, note that by construction we have

$$q_i^T := \sum_{X' \in \Lambda_i^S(X)} q^T_{X,X'} = \tau \sum_{X' \in \Lambda_i^S(X)} p^T_{X,X'} = \tau.$$

Therefore, the rate at which each site i transitions is common to all sites. This kind of Markov chain is called a random sweep Gibbs sampler. Peers stay at each configuration an exponential amount of time, of parameter τ, after which they choose a new configuration $X' \in \Lambda_i^S(X)$ with probability $p^T_{X,X'}$.

Remark 3: When the temperature T goes to zero, the transitions of peer i are dominated by configurations that minimize the *local* energy $\mathcal{E}_i$; as we saw in the previous section, these correspond to a tit-for-tat rule unchoking peers from whom it is currently downloading the fastest, similar to BitTorrent. The difference between this algorithm and BitTorrent lies in the manner that we introduce its randomness. Instead of having always an optimistic connection that blindly explores another peering options, this algorithm chooses all of its connections using the same distribution. If at some point we reach a state with local energy close to zero (e.g. when the peer is exchanging with other peers with the same upload capacity), the probability of choosing a different peer is very small, making the current configuration stable. This is the key to obtaining an allocation as close as possible to proportional fairness, while retaining the capability of random search.

B. Systematic sweep Gibbs sampler

An alternative to the random sweep Gibbs sampler is the *systematic sweep Gibbs sampler*, in which each site is updated in a particular deterministic order, multiplying the transition probabilities of each row as the sequence goes along. It is most convenient here to define a discrete-time Markov chain that tracks the configuration state after each full sweep, with transition probabilities $p^T_{X,X'}$ now involving changes in all matrix rows, with the following form:

$$p^T_{X,X'} = \prod_{i=1}^N \frac{e^{-\frac{1}{T}\mathcal{E}_i(X,X')}}{Z_i^T},$$

where

$$\mathcal{E}_i(X, X') = \sum_{j=1}^{i-1} (\mu_i x'_{ij} - \mu_j x'_{ji})^2 + \sum_{j=i}^{N} (\mu_i x'_{ij} - \mu_j x_{ji})^2,$$

and Z_i^T are appropriate normalizing constants. $\mathcal{E}_i(X, X')$ reflects the local energy of the i-th intermediate configuration when transitioning between X and X'.

The Markov chain defined before has a finite state space and is irreducible and aperiodic, thus it eventually converges to its invariant distribution, which can be shown to be equal to π^T. Furthermore, in this case we can bound the speed of convergence. We state the following result, but the proof is omitted due to space limitations.

Proposition 10: Let η be the initial distribution and P be the transition matrix of the discrete-time Markov chain. Denote by $d_V(\cdot,\cdot)$ the total variation distance between two probability measures. If the network is full mesh, then

$$d_V(\eta P^n, \pi_T) \leq \frac{1}{2} d_V(\eta, \pi_T) \left(1 - e^{-\sum_i \frac{2\mu_i \mu_{\max}}{T N_0}}\right)^n.$$

The systematic sweep sampler would correspond in practice to the case where each peer updates its connections after a fixed amount of time. this is the version that we chose to implement in the simulations below.

V. Implementation and Simulations

We now evaluate the devised systematic sweep Gibbs algorithm as a means to achieve reciprocity and fairness. We implemented the algorithm in Matlab, and in order to perform comparisons, we also implemented idealized versions of the BitTorrent unchoking mechanism, as well as the ideal proportional reciprocity based on the Sinkhorn iteration discussed in Section II, the PropShare unchoking algorithm of [9] and the Markov approximation approach devised in [12].

Let us begin by briefly recalling the different algorithms. The standard BitTorrent unchoking mechanism maintains for each peer $N_0 = 4$ outgoing connections. Three of these connections are used to reciprocate other peers, and the remaining connection is an *optimistic* unchoke, designed to explore new peers. The latter is kept for several iterations in order to allow time for the optimistically unchoked peer to reciprocate. This algorithm has low overhead and enables peers to find appropriate partners [7], but it has two main disadvantages: the unchokes are based only on the ranking of better contributors, and not in the bandwidth they provided, which has incentives problems [9]. It also constantly searches for new peers, allocating a substantial proportion of the uplink bandwidth to this end, and possibly drifting away from good configurations.

The Sinkhorn algorithm, on the other hand, focuses on reciprocating only, by allocating proportional shares to each connected peer. To this end, it is the best one can do and achieves a fast convergence time. A pure proportional response however, has two main drawbacks from a practical perspective: it requires to keep a large amount of connections with several peers, as well as controlling exactly the amount of bandwidth allocated to each unchoked peer, which may be difficult to implement in practice. More importantly, it can get stuck in bad configurations if the initial connectivity of peers is sparse.

The PropShare algorithm is based on the Sinkhorn iteration, and was devised to correct this last problem, among other optimizations. This algorithm allocates proportionally to the received contributions 80% of the uplink bandwidth of a given peer. It uses the remaining 20% to explore new peers through optimistic unchokes, much like BitTorrent. This exploration mechanism enables the algorithm to increase the number of connected peers. While this may achieve a higher level of fairness, the bandwidth committed to the optimistic search can make the algorithm drift away from good configurations. This algorithm still suffers from the burden of maintaining many connections and controlling the amount of bandwidth given to each. In our simulations we implemented an idealized version of PropShare based on these features, not taking into account these problems, and thus we expect our results to provide an upper bound on real life PropShare performance.

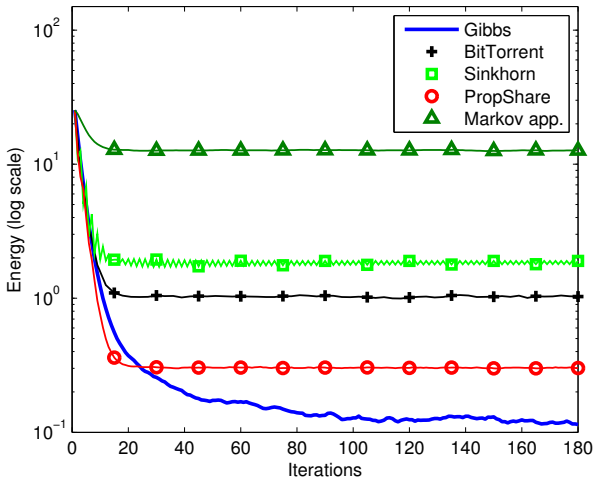

Fig. 1. Gibbs energy evolution for the different algorithms.

Finally, the Markov approximation algorithm of [12] has points in common with the one proposed in this paper. However, the main emphasis is on achieving optimal throughput allocation by discovering the best neighbors to *upload* to. Reciprocity is not taken into account and therefore it suffers on the fairness side, as we will show.

To evaluate the algorithms, we simulated a scenario with $N = 100$ peers, which belong to two categories: half of the peers have a fast uplink connection, and the other half are 5 times slower. Ideally, all peers should get as much bandwidth as they give to achieve proportional fairness. All peers can potentially connect to each other, and in the case of Gibbs and BitTorrent, dividing bandwidth equally between all outgoing connections. All algorithms start from the same initial connectivity condition with $N_0 = 4$ outgoing connections per peer.

As a measure of the achieved reciprocity and fairness, we evaluate two metrics: the Gibbs energy $\mathcal{E}(X)$ from (3) defined in Section III, which is intended to be minimized by the Gibbs algorithm, and also the Kullback-Leibler (KL) divergence $D(\mu\|r)$ between the offered bandwidths μ_i and the rates received by each peer r_i. Recall from Section II that the KL divergence is related to a weighted proportional

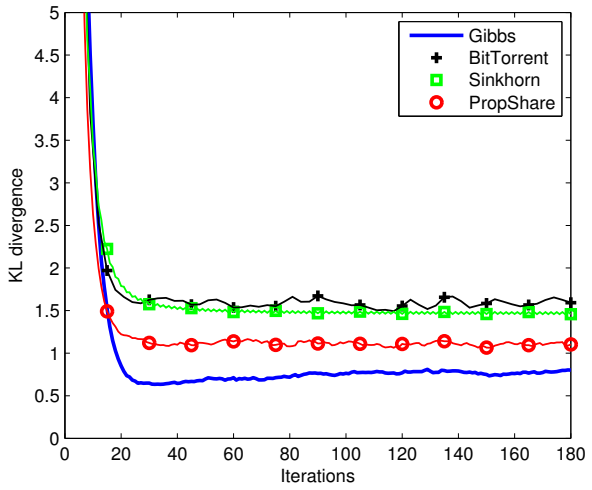

Fig. 2. Evolution of the Kullback-Leibler divergence between uplink and received rate.

fairness criterion, and in the optimal allocation $D(\mu\|r^*) = 0$.

In order to correctly assess the performance of each algorithm, we simulate several replications of each one of them starting from a random initial condition, and plot the average results for each metric. In Fig. 1, we plot the evolution of the Gibbs energy $\mathcal{E}(x)$ for the different algorithms in log scale.

The Gibbs algorithm (with $N_0 = 4$ in this case for comparison) is designed to find a minimum of the energy and it does so in a competitive number of iterations, at the expense of a convergence time somewhat slower than the remaining algorithms. The Markov approximation algorithm is not good at achieving reciprocity, remaining with a high energy. The (ideal) Sinkhorn iteration is theoretically the best, but when facing random initial conditions with sparse connectivity the algorithm cannot fully renormalize the allocation, and thus it does not reach minimum energy. The BitTorrent algorithm is assigning too many optimistic unchokes, and this reflects on the energy achieved. Finally, PropShare is the best alternative, at the expense of having a greater number of simultaneous connections for each peer, and controlling bandwidth on each one of them. The Gibbs algorithm achieves a better reciprocity while at the same time having only $N_0 = 4$ open connections per peer sharing the uplink rate equally.

As for the fairness in the resulting allocation, in Fig. 2 we plot the aforementioned KL divergence for each algorithm. In this case we omit the Markov approximation algorithm since it does not pursue proportional fairness. Note that also for this metric the best algorithms are the proposed Gibbs sampler and the PropShare algorithm, with the Gibbs sampler achieving better results.

VI. Conclusions

In this paper we discussed the resource allocation of P2P networks under heterogeneity in access bandwidth. We reviewed the proportional response dynamics and its connection with the Sinkhorn iteration, remarking the benefits of this allocation over the resulting one with BitTorrent through its incentives. We proposed a decentralized algorithm based on a Gibbs sampler that approximates the proportional allocation while being easier to implement than the proportional response dynamics, reaching a middle ground between the simplicity of BitTorrent and the fairness and incentives of the proportional response. Moreover, we explored through simulations how the new algorithm compares to several alternatives in an heterogeneous P2P environment, showing good results.

References

[1] C. Aperjis, R. Johari, and M. J. Freedman, "Bilateral and Multilateral Exchanges for Peer-Assisted Content Distribution," *IEEE/ACM Transactions on Networking*, vol. 19, no. 5, pp. 1290–1303, 2011.

[2] H. Balakrishnan, I. Hwang, and C. J. Tomlin, "Polynomial Approximation Algorithm for Belief Matrix Maintenance in Identity Management," in *Proc. of IEEE CDC*, 2004.

[3] S. Boyd and L. Vanderberghe, *Convex Optimization*. Cambridge, UK: Cambridge University Press, 2004.

[4] P. Brémaud, *Markov Chains: Gibbs Fields, Monte Carlo simulation and queues*. Springer, 1999.

[5] B. Cohen, "Incentives build robustness in BitTorrent," in *Proc. of 1st Workshop on the Economics of Peer-2-Peer Systems*, 2003.

[6] P. Erdős and T. Gallai, "Graphs with prescribed degrees of vertices (Hungarian)." *Mat. Lapok*, vol. 11, pp. 264–274, 1960.

[7] B. Fan, J. C. Lui, and D.-M. Chiu, "The Design Trade-offs of BitTorrent-like File Sharing Protocols," *IEEE/ACM Transactions on Networking*, vol. 17, pp. 365–376, 2009.

[8] F. Kelly, *Reversibility and stochastic networks*. Wiley, 1979.

[9] D. Levin, K. LaCurts, N. Spring, and B. Bhattacharjee, "BitTorrent is an Auction: Analyzing and Improving BitTorrent's Incentives," in *Proc. of ACM SIGCOMM*, 2008.

[10] R. T. Ma, S. C. Lee, J. C. Lui, and D. K. Yau, "Incentive and Service Differentiation in P2P Networks: A Game Theoretic Approach," *IEEE/ACM Transactions on Networking*, vol. 14, no. 5, pp. 978–991, 2006.

[11] M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy, and A. Venkataramani, "Do incentives build robustness in BitTorrent?" in *Proc. of NSDI*, 2007.

[12] Z. Shao, H. Zhang, M. Chen, and K. Ramachandran, "Reverse-Engineering BitTorrent: A Markov Approximation Perspective," in *Proc. of IEEE INFOCOM*, 2012.

[13] R. Sinkhorn, "A relationship between arbitrary positive matrices and doubly stochastic matrices," *Annals of Mathematical statistics*, vol. 35, pp. 876–876, 1964.

[14] ——, "Diagonal equivalence to matrices with prescribed row and column sums II," *Proc. of the American Mathematical Society*, vol. 45, no. 2, pp. 195–198, 1974.

[15] R. Sinkhorn and P. Knopp, "Concerning nonnegative matrices and doubly stochastic matrices," *Pacific Journal of Mathematics*, vol. 21, no. 2, pp. 343–348, 1967.

[16] F. Wu and L. Zhang, "Proportional response dynamics leads to market equilibrium," in *Proc. of ACM STOC*, 2007.

[17] X. Yang and G. de Veciana, "Performance of peer-to-peer networks: Service capacity and role of resource sharing policies," *Performance evaluation*, vol. 63, pp. 175–194, 2006.

[18] H. Zhang, Z. Shao, M. Chen, and K. Ramachandran, "Optimal Neighbor selection in BitTorrent-like Peer-to-Peer Networks," in *Proc. of ACM SIGMETRICS*, 2011.