Fairness, incentives and performance in peer-to-peer networks

Gustavo de Veciana and Xiangying Yang

Department of Electrical and Computer Engineering
The University of Texas at Austin
Austin, TX 78712
{gustavo,yangxy}@ece.utexas.edu

Abstract

In this paper we consider the sensitivity of the 'service capacity' of peer to peer (P2P) file sharing applications to demands, user cooperation, and providing incentives for cooperation through fairness guarantees. To do so we propose models for both the steady state performance and the transient characteristics of the system. Both are of interest since, from the perspective of a given file, demands are prone to be initially bursty, e.g., flash crowds, and subsequently be roughly 'stationary.' We will show that P2P systems can be exceptionally responsive to bursty demands by increasing their service capacity exponentially, while exhibiting nice performance scaling to the offered loads during stationary regimes. We also study system characteristics associated with various fairness criteria. We provide partial validation of our models and conclusions based on traces obtained from a second generation P2P application.

1 Introduction

Peer-to-peer (P2P) architectures for file sharing among ad hoc, possibly dynamic, collections of hosts are generating an increasing fraction of the traffic on today's Internet and are reshaping the way new network applications are designed. The idea is to have hosts participate in an application level overlay network enabling signaling, routing, and searching among participating hosts. Once a host locates the document(s) of interest, direct connections are established to mediate their transfer. The key principle is to allow, and in fact encourage, participating hosts to play dual roles as servers and clients. Using similar ideas researchers are pursuing work on "grid computing" to enable not only file sharing, but distributed content delivery, storage, and computation over overlay networks, see e.g., [1] [2]. Our goal in this paper is to analyze performance aspects for these types of networks.

What is the service capacity of a P2P System? In evaluating the 'service capacity' of a P2P system it makes sense to consider both the transient and stationary regimes. The system might enter a transient regime, in response to a large burst of requests for a popular file, perhaps when it is first introduced. During such periods relatively small numbers of peers may initially have copies of the file to serve others. However, as replication proceeds the large number of peers requesting the document can be leveraged as servers enabling an exponential growth in the service capacity of the system, until the burst of requests is served. Once the intensity of requests stabilizes, the system enters a 'steady state' where the throughput performance of each peer is stable. These two phases are exhibited in the representative trace for the average throughput seen by peers in the system as a function of time shown in on the left in Fig.1. This trace was obtained by monitoring the BitTorrent P2P system [3]. The trace begins with the addition of a new document to a P2P system, then, the solid line tracks an exponential growth in service capacity corresponding to a transient period, and finally the dotted line corresponds to fluctuations in a steady state. Note that during the 'steady state' the request rate is only approximately stationary. Indeed, not shown in Fig.1, the demand rate is fluctuating yet the average performance per peer is fairly stable. As will discussed in the sequel, during the steady state period the service capacity tends to scale with the demand. This example exhibits a desirable exponential growth, and subsequent self-scaling (based on popularity) of a P2P system's service capacity for a given document.

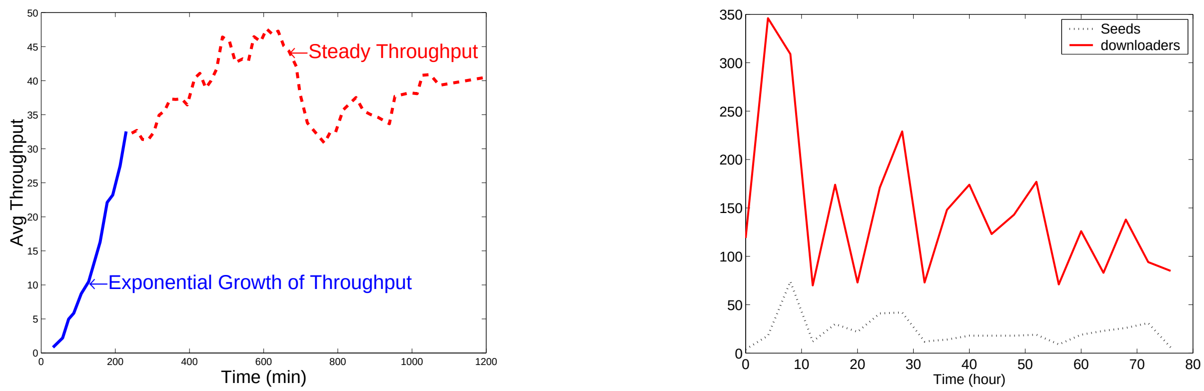

Figure 1: Two representative traces for the BitTorrent system. On the left two-phases in the evolution of the average throughput per peer versus time after a single document introduced into a P2P network. On the right a trace exhibiting a significant amount of variability in the number of servers (seeds) and downloaders as a function of time.

The service capacity in these two regimes depends on a number of factors:

- data management: a document may be partitioned into various parts permitting concurrent downloading from multiple peers; the granularity and placement of these is critical;
- peer selection: the mechanism whereby a peer is selected as a server may take into account load balancing, bandwidth availability, and differentiate among peers who contribute more to the community;
- admission, scheduling and/or bandwidth allocation policy: limiting the number of concurrent downloaders and/or priority scheduling of service or differentiation in the bandwidth allocated among peers;
- traffic dynamics: the request processes for documents along with the dynamics of how peers stay online and/or delete documents.

These factors are interrelated in complex ways. For example, a good peer selection scheme may favor peers that are likely to subsequently stay as servers for the document and thus contribute to the system's service capacity. Multi-part downloads can increase the rate at which files get duplicated while at the same time allowing users to serve as peers for parts they have already obtained prior to completing downloads. Allowing large numbers of peers to download from one another may increase the subsequent potential service capacity for a document but may increase delays. Spatial clustering of peers may impact the service capacity of a P2P system in subtle ways, since serving a peer which is far away and may have low bandwidth, may subsequently help to quickly serve an entire neighborhood of interested peers. Recognizing some of these relationships new P2P applications attempt to use *credit based systems* to provide incentives for peers to stay online and 'increase' their upload bandwidth to serve other peers [3]. This is often done by keeping peers' credit history and based on their contributions (e.g., upload volume) give them different priority in transfers or access resources on other peers. Such mechanisms are geared at providing incentives for peers to cooperate. As we will see in the sequel their impact on performance may be subtle and significant. Our goal in this paper is to explore the interactions among some of these factors and their impact on performance from the perspective of a P2P system's transient and stationary service capacity.

Related Work. Most research on P2P systems so far has emphasized design, traffic measurement and workload analysis but not performance evaluation. Early work by [4][5][6] studied traces of P2P applications like Gnutella and Napster. They focused on characterizing the overall P2P system, e.g., request patterns, traffic volume, traffic categorization and properties of shared online content as well P2P structure and dynamics, e.g., connectivity and peer/host behaviors. More recent research in the direction of evaluating P2P systems has focused on performance. Peer selection schemes were evaluated in [7], where measurements are used to optimize the selection of good peers and improve the overall system performance. A few researchers have used analytical models to study the performance of P2P networks. For example, [8] constructed a model for the signaling messages in the Gnutella network and concluded that signaling might significantly compromise performance. The work in [9] is among the first to model a general P2P system and evaluate its performance. Their model, a closed queuing system, provides basic insights on the stationary performance of a P2P system; among these, the dependence of performance on parameters like the peer request rate and number of peers in the system.

Organization of paper. The rest of the paper is organized as follows. In §2 we discuss and analyze the performance of a P2P system from the point of view of the transient and stationary regimes. We show some performance scalability properties for such systems when they are subject to large transient or stationary loads as well as performance sensitivity to user behavior characteristics, e.g., cooperativeness and heterogeneity. With a view on providing users incentives to cooperate in §3 we consider the role that fairness in bandwidth allocation can play in P2P systems. We conclude our work in §4.

2 Service Capacity of P2P systems

Transient regime –branching process model. We consider the transient regime to be associated with a large, say n, burst of roughly concurrent requests in a P2P system when there are initially limited numbers of peers, say 1, able and willing to serve them. As mentioned earlier this may arise when a document is first introduced. For some types of files, e.g., those whose interest is highly dependent on timeliness, the majority of requests for such a document may occur during such bursts or may occur during periodic (daily) cycles, and thus the performance seen by users is dominated by that seen in the transient regime –see right panel in Fig. 1 for an example of a trace exhibiting this type of behavior.

The underlying file sharing mechanism in the transient regime is best explained based on a deterministic model. Suppose that $n = 2^k$ users wish to acquire a document which is initially available at one peer. Assume that each peer has a limited upload capacity, say b bps, and network capacity is otherwise unconstrained, i.e., download capacity is assumed to be unconstrained.

Suppose the document has size s bits, and a peer can only serve a document once it has been fully downloaded. Thus to serve n requests ns bits will need to be exchanged. It should

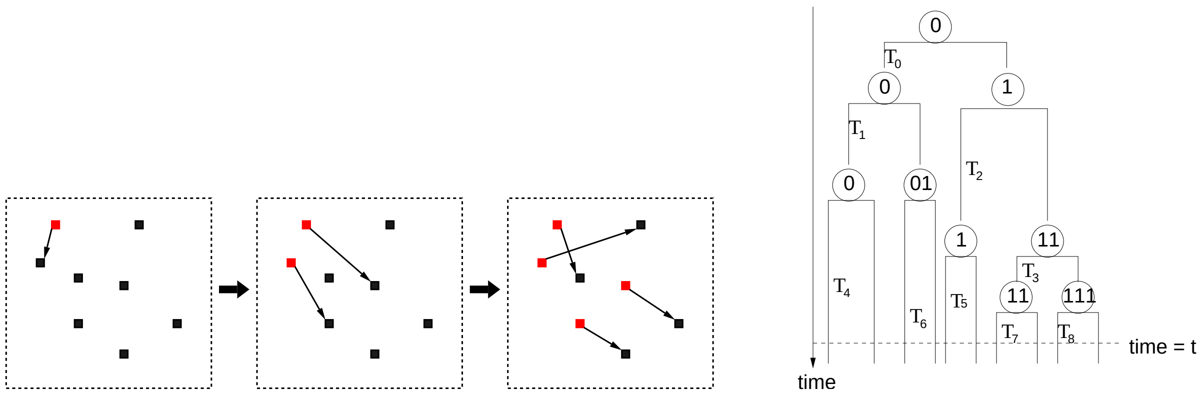

Figure 2: On the left file sharing in a P2P system. On the right a branching process model for file replication.

be clear that a good strategy is to first serve one user at rate b, at which point the service capacity grows to $2b$, and then have these two peers serve additional users, until the n users are served. As shown in Fig.2, under this idealized strategy peers will complete service every $\tau = s/b$ seconds, at which point the number of peers that can serve the document doubles, leading to an exponential growth of $2^{t/\tau}$ in the number of peers available to serve the document. If the system follows these dynamics the n peers will be served by time $\tau \log_2 n = \tau k$. Thus the 'average' download delay $\bar{d}$ experienced by peers can be computed as follows. Let d_j denote the delay experienced by the jth peer to complete, and note that $2^{i-k}n$ peers complete service at time $(i+1)\tau$, giving an 'average' delay for peers in this transient regime of:

$$\bar{d} = \frac{1}{n}\sum_{j=1}^{n} d_j = \sum_{i=0}^{k-1} 2^{i-k}\tau(i+1) = k\tau - \frac{n-1}{n}\tau = \tau\big(\log_2 n - \frac{n-1}{n}\big) \approx \tau \log_2 n.$$

Hence although the system sees an initial burst of n requests the average delay seen by peers scales as $\log_2 n$ which is favorable relative to the linear scaling of n one would obtain for a system with a fixed set of servers. A typical technique to further speed up such systems is to enable multi-part downloads. For example, suppose the file is divided into m chunks with identical size, and peers can serve such chunks immediately once they are downloaded. This permits pipelining file transmissions, and can be shown to achieve a further reduction in delay by a factor of $1/m$.

The above idealized deterministic model exhibits the basic transient dynamics for capacity and performance one might expect from a properly designed P2P system. However we assumed service times were deterministic and peers cooperative to the extent that whenever they complete downloading a document (chunk) they will turn around and serve it. In general service times may exhibit variability, due to congestion or heterogeneity in the capacity of peers, and a fraction of the peers may choose not to cooperate by leaving the system.

In order to capture such variability one might devise a branching process model for the system. Let $N(t)$ denote the number of peers available to serve a given document at time t. Assume that initially there is but one copy of the document in the network, i.e., $N(0) = 1$ with probability 1, and a large number of interested peers.

Fig. 2 shows a typical evolution of the file sharing process assuming each peer serves one other peer at a time. Thus, initially Peer 0 shares its file with Peer 1. After a random service time T_0, this process completes, and Peers 0 and 1 can now serve other peers. As shown in the figure Peer 01 and Peer 11 now download from Peer 0 and Peer 1 respectively and complete this process after some random times T_1 and T_2 respectively. This replication process continues to evolve over time, as long as there are peers still requesting the document. Suppose the times to realize a transfer between peers $T_i,\ i = 0,1,\ldots$ can be modeled as independent random variables with a common distribution, i.e., $T_i \sim T$ where $F_T(t) = P(T \leq t)$ and $E[T] = \tau$. This distribution captures the variability in the transfer time at congested or heterogenous peers, while the mean reflects the typical upload and/or download performance among peers in the system.

The model we have described corresponds to a standard age-dependent branching process with a fixed family size $v = 2$ at each new generation. General results for the evolution of the mean population, i.e., service capacity of our P2P model, can be found in [10] §10.4.22 and [11] Chap. IV Thm.3A. The following is a restatement of the basic result: for a branching process with i.i.d. family sizes having the distribution of a integer valued random variable V, with $E[V] = v$.

Theorem 1 *In the super critical case where the mean family size per generation satisfies $E[V] = v > 1$ and F_T is non-lattice, the expected population of an age dependent branching process for large t is given by*

$$E[N(t)] \sim \delta e^{\beta t}, \tag{1}$$

where $\beta > 0$ is such that $d\tilde{F}(x) = ve^{-\beta x} dF_T(x)$ is a probability distribution function, i.e., $\int_0^\infty ve^{-\beta x} f_T(x)dx = 1$ whose mean we denote by $\tilde{\tau}$ and where $\delta = \frac{v-1}{\tilde{\tau}\beta v}$.

Thus for the P2P model in Fig.2 the mean service capacity grows exponentially with β and δ as defined in Thm.1 and where $v = 2$.

We can use this model to crudely examine the impact of variability and design choices on the transient capacity of the system:

Variability in generation times improves the growth exponent. Given two distributions for the generation times associated with the random variables T^1 and T^2 such that $E[T^1] = E[T^2]$ and there is an increasing convex ordering (see [12]) $T^{(1)} \leq^{icx} T^{(2)}$ one can show that the growth exponents satisfy $\beta^1 < \beta^2$.

Increased parallelism typically may decrease the growth exponent. Consider the branching process model where each peer serves $(v-1) > 1$ requests in parallel. As a crude model for the delays under such sharing let us assume a slow down in the generation time of $(v-1)$. If we consider a generation time distribution defined on $[c,\infty)$, where $c > 0$, and it has tail decaying as or faster than exponentially or has a finite mean τ, one can show that the growth rate for the branching process decreases roughly as $\frac{\ln(v)}{(v-1)\tau}$, where $\tau > \bar{\tau} > c$.

Uncooperative peers and parallel downloads. Suppose that upon completing a download, a peer deletes the file and or leaves the system with probability ζ. In this case one must ensure that $v\zeta > 1$ to maintain the branching process in the super critical regime. One can further show that when peers exhibit such uncooperative behavior it may be worthwhile to have higher parallelism, i.e., higher v in order to keep the growth rates high.

In summary, the transient regime for a P2P system exhibits excellent scalability properties, in the sense that average delays for service grow logarithmically in the burst size. In addition further speedups can be obtained by enabling multi-part file sharing. In order optimize the growth rate for the transient regime it may be worthwhile to have peers serve others in parallel to ensure a sufficient peers survive each generation.

Stationary regime –Markov chain model. For some types of files, after an initial transient phase, the demand may be fairly stationary and sustained. If this is the case it makes sense to consider performance as driven by that in stationary regimes. Below we briefly describe a model for such a regime and some empirical results.

Consider all peers in a P2P system which are interested in, or serving, a particular document and assume that there will always be at least one peer serving the document. Suppose new requests follow a Poisson process with rate λ. The system's state is a pair $(x,y) \in \mathbf{N} \times \mathbf{N}^+$, where x denotes the number of peer requests currently in progress or queued and y denotes the number of peers that have finished downloading and still remain in the system, i.e., contributing to the system's service capacity. We further assume that the file is partitioned into chunks, allowing multi-part downloading, thus peers which are in the process of downloading, but already have part of a file, can serve other peers. Suppose then that downloading peers contribute to the system's service capacity, but their contribution is only a fraction η of that of a peer who has already downloaded the full document. The total service capacity of the system is thus proportional to the effective number of servers in the system, we denote it by $\mu(\eta x + y)$, where μ is the service rate for a request at a peer which can serve the document in full. Each time a peer completes downloading the document it becomes a server in the system, but each such peer may leave the system at rate γ. Thus in this model the service time for a request at a single peer and time until peer which has completed a download leaves the system are independent and exponentially distributed with rates μ and γ. The evolution for the state of this system can be described by a continuous time Markov chain with a rate transition matrix Q over the state space $\mathbf{N} \times \mathbf{N}^+$ given by :

$$\begin{array}{lll} q\big((x,y),(x+1,y)\big) & = \lambda & \text{new request} \\ q\big((x,y),(x-1,y+1)\big) & = \mu(\eta x + y) & \text{service a peer} \\ q\big((x,y),(x,y-1)\big) & = \gamma y & \text{exit system.} \end{array}$$

We numerically computed the stationary distribution for this Markov chain to find mean number of jobs, servers, and delay for this system. The left panel in Fig. 3 exhibits the average delay in the system for a range of parameters; specifically $\mu = 4.0$ $\eta = 0.5$ and varied the values of λ and γ from 4.0 to 12.0 and 2.0 to 8.0, respectively. The average delay depends only on the ratios $\frac{\lambda}{\mu}$, the offered load and $\frac{\gamma}{\mu}$, the rate at which peers exit the system, as long as delays are measured in the units of holding times μ^{-1}. As can be seen the mean delay seen by peers in this system model may increase or decrease with the offered load depending on the rate γ/μ at which peers exit. Not shown in the figure, is the fact that this threshold depends on the effectiveness η of document sharing in the P2P system.

Although this is a simple model, we expect similar performance characteristics in real systems. The data shown in Fig.3 corresponds to a sample of 500 files with file sizes ranging from 400 MBytes to 1.1 GBytes, for which the system capacity appeared to be in the steady state, i.e., one to four days have elapsed since these documents were introduced to the system and the service capacity and throughput per peer should be representative of their popularity/offered loads. For each file, we plot the KByte transmission delay, i.e., inverse of the average throughput per peer (in KByte/sec), versus the number of servers and downloaders participating in the system. The number of participants is roughly linear in the offered load for a each file, i.e., a proxy for the popularity of the document. For files with less than 50 peers participating in the system, i.e., not very popular, the performance is seen to be quite unpredictable. Intuitively, this big variance is due to the fact that the number of peers is small and heterogeneity among peers is reflected in differences in performance. However, for files that are very popular, the performance improves, albeit slowly, in the number of participants. This matches our analytical

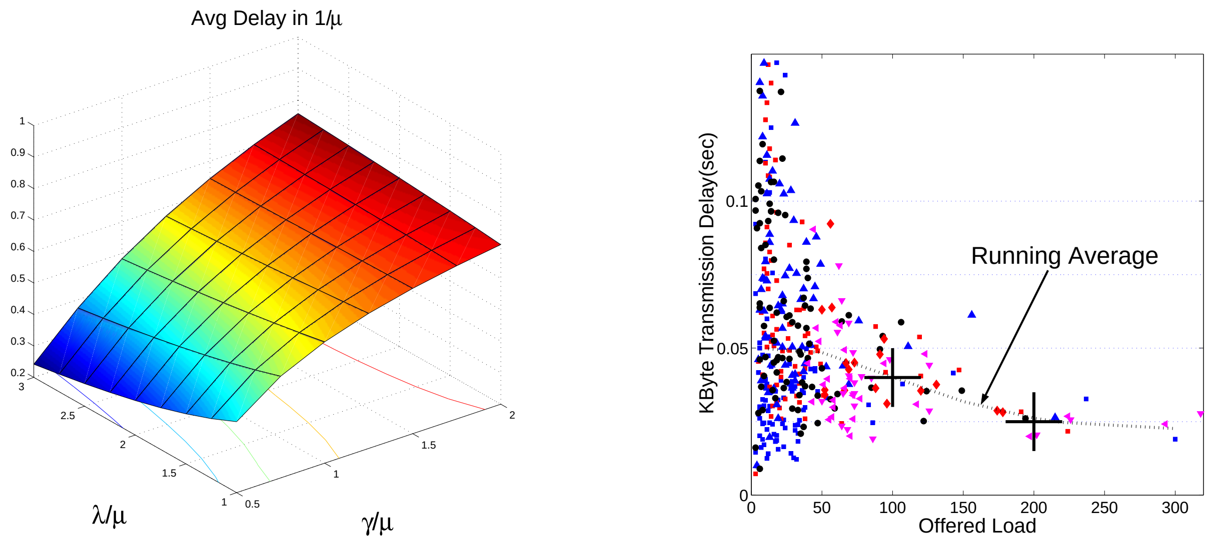

Figure 3: On the left the average delay for the Markov model with $\eta = 0.5$ for a range of system parameters. On the right the KByte transmission delay versus offered load(estimated by summing over servers and downloaders associated with a given file in the BitTorrent P2P system.

results very well, i.e., average delays appear to go down in the 'offered load'.

Our empirical data and system model suggest that a P2P system in steady state will likely exhibit fairly good performance. As is the case in the transient regime a significant amount of the service capacity is leveraged from peers that are concurrently downloading the file rather than peers that have already obtained it in full. It is, however, useful to provide incentives for the latter to stay in the system. Indeed we note that if the rate at which peers leave the system is slow, then documents with a high offered load may see improved average performance versus those with lower loads. This self-scaling characteristic would in practice be highly desirable since it achieves better performance for under higher offered loads.

3 Fairness and incentives in P2P systems.

One may consider introducing fairness criteria in P2P systems as a means to provide users incentives to collaborate, i.e., encouraged them to contribute, by serving documents that are in demand, increase their upload bandwidth, or stay in the system longer.

Consider the following model for these types of service/bandwidth allocation. We let N denote a set of peers and and $\mathbf{x} = (x_{ij} | i \neq j, i, j \in N)$ denote the service provided by each peer, i.e., x_{ij} is a measure of the allocation of resources from i to j at time t. Finally for simplicity let us assume that the access rates are such that each peer is not download constrained but has an upload constraint captured by $\mathbf{b} = (b_i | i \in N)$ where b_i denotes the upload bandwidth constraint for peer i. This may be a constraint placed by the communication system or an artificial one placed by the user. Below we will assume that that peers are fully connected and share a common interest, i.e., have demands from each other. We let the set N_i denote the neighbors for peer i, which in the case of fully connected nodes would be given by $N_i = N \setminus \{i\}$.

First reasonable notion of fairness is an bandwidth allocation whereby the service a given peer receives from another is proportional to its own overall contribution to the system. We shall say an allocation of service capacity $\mathbf{x}$ is said to be *globally proportionally fair* if for all

$i,j \in N$ where $i \neq j$ it satisfies

$$x_{ij} = \frac{u_j}{\sum_{k \in N_i} u_k} b_i \text{ where } u_j = \sum_{i \in N_j} x_{ji}$$

the idea is to allocate outgoing bandwidth from node i to a peer j in proportion to its overall contribution to the system u_j. If there is demand from each peer, and there are no binding download constraints under this policy $u_j = b_j$ and it follows that

$$x_{ij} = \frac{b_j}{\sum_{k \in N_i} b_k} b_i, \text{ and } d_j = \sum_{i \in N_j} x_{ij} = b_j \sum_{i \in N_j} \left[\frac{b_i}{\sum_{k \in N_i} b_k} \right].$$

Here d_j denotes the *aggregate download bandwidth* that j would realize. Unfortunately this criterion, while appealing, requires tracking of overall contributions of each nodes, and is not amenable to distributed implementation.

An alternative is for a peer to allocate its upload capacity to other peers based directly on the service it has received from them. This would only require peers to keep track of those peers which have served them, which is fairly straightforward to do and verify by peers in a distributed manner. We shall say that an allocation is *peer wise proportionally fair* if for all $i,j \in N$ where $i \neq j$ it satisfies

$$x_{ij} = \frac{x_{ji}}{\sum_{k \in N_i} x_{ki}} b_i.$$

In general such allocations are not unique and may be, perhaps unexpectedly unfair. The following theorem summarizes some of the characteristics of such allocations when they are strictly a positive.

Theorem 2 *Under a peer wise proportionally fair allocation which is strictly positive i.e., $x_{ij} > 0, \forall i \neq j$, the bandwidth allocation must be symmetric $x_{ij} = x_{ji}$. Furthermore suppose that $|N| = n$ peers are indexed such that their upload bandwidths are nondecreasing, i.e., $b_1 \leq b_2 \ldots \leq b_{n-1} \leq b_n$, then a necessary and sufficient condition for the existence of a convex set of strictly positive peer wise proportionally fair allocation is that*

$$b_n < \sum_{i=1}^{n-1} b_i \tag{2}$$

We shall to (2) as the non-dominant condition.

By the symmetry property, it follows that a strictly positive peer wise proportionally fair bandwidth allocation policy, will satisfy $u_i = d_i = b_i$, i.e., a peer's download bandwidth is equal to the upload bandwidth it contributes to the system.

Consider 3-node case and suppose the non-dominant condition is satisfied then we have that

$$b_1 + b_2 > b_3, \; b_1 + b_3 > b_2, \text{ and } b_2 + b_3 > b_1.$$

For this case there is a *unique* strictly positive allocation satisfying the peer-wise fairness policy:

$$\mathbf{x} = [x_{ij}] = \begin{bmatrix} 0 & \frac{b_1+b_2-b_3}{2} & \frac{b_1+b_3-b_2}{2} \\ \frac{b_2+b_1-b_3}{2} & 0 & \frac{b_2+b_3-b_1}{2} \\ \frac{b_3+b_1-b_2}{2} & \frac{b_3+b_2-b_1}{2} & 0 \end{bmatrix}$$

The uniqueness however does not follow when there are more than three peers. Instead as stated in Thm. 2 strictly positive solutions belong to a convex set of symmetric solutions. There may also be solutions that are not strictly positive. In this case the symmetry need not hold, and a peer's download allocation need not be equal to its upload contributions. For example, in the 3-node case, the following allocations are feasible:

$$\mathbf{x} = \begin{bmatrix} 0 & \frac{b_1 b_2}{b_2+b_3} & \frac{b_1 b_3}{b_2+b_3} \\ b_2 & 0 & 0 \\ b_3 & 0 & 0 \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} 0 & b_1 & 0 \\ \frac{b_2 b_1}{b_1+b_3} & 0 & \frac{b_2 b_3}{b_1+b_3} \\ 0 & b_3 & 0 \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} 0 & 0 & b_1 \\ 0 & 0 & b_2 \\ \frac{b_3 b_1}{b_1+b_2} & \frac{b_3 b_2}{b_1+b_2} & 0 \end{bmatrix}$$

Such solutions may not desirable. Indeed they do not really guarantee a natural notion of fairness. For example, it is likely that a node contributing a large total upload bandwidth only gets rewarded by small download throughput from peers with low upload bandwidths. For example, consider the first allocation above where $u_1 \gg u_2, u_3$. Second, such allocation may lead to disconnected cliques of peers. This in turn would have a negative impact on the performance since the ability to leverage distributed service capacity would be reduced, and peer may in practice have trouble finding a file/chunk it needs.

To avoid these problems it is preferable to compromise the notion of peer wise proportional fairness to ensure that there exists a unique positive solution and distributed mechanisms to reach it. A simple idea is to ensure that peers persistently attempt exchanges with others thus avoiding becoming disconnected. We can model this by ensuring there is a minimal exchange volume among all peers, say ε. With this in effect we define a a bandwidth allocation as being ε-*peer wise proportionally fair* if for all $i, j \in N$ where $i \neq j$ it satisfies

$$x_{i,j} = \frac{x_{j,i} + \varepsilon}{\sum_{k \in N_i} (x_{k,i} + \varepsilon)} b_i.$$

First, note that for ε small enough, one would expect the resulting allocation to be close to one which is strictly positive and satisfies the peer wise proportionally fair requirement, when one exists. If however ε were large a peer would allocate its upload bandwidth equally among its neighbors. Second, a peer's download throughput d_i increases with its upload bandwidth b_i as is the case for the globally proportionally fair allocation. Third, depending on ε bandwidth allocations are biased, in that a peer with limited upload bandwidth is likely to get a higher relative aggregate download throughput versus one with a higher upload bandwidth.

4 Conclusions and Future work

In this paper we propose and discuss several models for P2P file sharing systems and show, perhaps not unexpectedly, that in both the transient and stationary regimes the degree to which users cooperate plays a critical role in determining the 'service capacity.' Based on, still limited, empirical data we argue that both transient and stationary regimes are significant in such systems. Indeed in some cases the offered load exhibits a substantial burst or periodic bursts of requests while in others the system is subject to a fairly stationary request pattern. As such performance optimization and system design should target improving the characteristics in both of these regimes.

Since the performance is sensitive to the degree of user cooperation it makes sense to provide incentives to users to share their resources. In particular it makes sense to increase their download allocations in a manner that depends on their contributions. We have presented a simple model for this problem and shown that an apparently simple criterion, peer wise proportionally fair bandwidth allocation, that might be implemented in a distributed fashion is problematic, in that it has a plurality of solutions some of which exhibit characteristics that are not fair at all, or shut of peers from other subsets of peers. To resolve this problem we propose an alternative criterion that we believe is amenable to distributed implementation yet will realize allocations that a roughly fair. In future work we will show that providing incentives to users so as to optimize performance in *both* the transient and stationary regimes may however subtle as one needs to balance fairness based on long term contributions of users versus leveraging the short term benefits of the resources offered by peers that are concurrently downloading a document during a transient burst.

References

[1] Sean Rhea, Chris Wells, and Patrick Eaton et al, "Maintenance-free global data storage," in *IEEE Internet Computing, pages 40–49*, September-October 2001.

[2] Sven Graupner, Winfried Kalfa, and Carsten Reimann, "Modeling and simulation of media-on-demand services - evaluating a digital media grid architecture," in *HP Laboratories technical report, HPL-2002-192*, 2002.

[3] Bram Cohen, "Incentives build robustness in bittorrent," in *URL http://bitconjurer.org/BitTorrent/bittorrentecon.pdf*, May 2003.

[4] Matei Ripeanu, Ian Foster, and Adriana Iamnitchi, "Mapping the gnutella network: Properties of large-scale peer-to-peer systems and implications for system design," *IEEE Internet Computing*, vol. 6, no. 1, pp. 50–57, Jan.-Feb.2002.

[5] Matei Ripeanu, "Peer-to-peer architecture case study: Gnutella network," in *Proceedings of First International Conference on Peer-to-Peer Computing, pages 99–100*, 2001.

[6] Stefan Saroiu, P. Krishna Gummadi, and Steven D. Gribble, "A measurement study of peer-to-peer file sharing systems," in *Proceedings of Multimedia Computing and Networking, San Jose*, January 2002.

[7] T.S. Eugene Ng, Yang hua Chu, Sanjay G. Rao, Kunwadee Sripanidkulchai, and Hui Zhang, "Measurement-based optimization techniques for bandwidth-demanding peer-to-peer systems," in *proceedings of IEEE INFOCOM03, San Francisco*, April 2003.

[8] Jordan Ritter, "Why gnutella can't scale. no, really," in *URL http://www.tch.org/gnutella.html*, 2001.

[9] Zihui Ge, Daniel R. Figueiredo, Sharad Jaiswal, Jim Kurose, and Don Towsley, "Modeling peer-peer file sharing systems," in *proceedings of IEEE INFOCOM03, San Francisco*, April 2003.

[10] G.R. Grimmett and D.R. Stirzaker, *Probability and Random Processes, 2nd Edition*, 1995.

[11] K.B.Athreya and P.E.Ney, *Branching Processes*, 1972.

[12] S.M. Ross, *Stochastic Processes*, 1983.