Bounds on the Performance of P2P Networks Using Tit-for-Tat Strategies

Dimitri do B. DeFigueiredo, Balaji Venkatachalam and S. Felix Wu
Department of Computer Science, UC Davis
{defigueiredo,bala,sfwu}@ucdavis.edu

Revision May 2, 2007

Abstract

Current P2P systems employ tit-for-tat strategies, where peers only upload when they are simultaneously downloading, to avoid free riding. We derive optimal tit-for-tat strategies and obtain theoretical bounds on the performance of any P2P network employing such strategies. These are fundamental limitations that stem from peers unwillingness to cooperate without getting something in return. We show that the number of cooperating peers in a tit-for-tat strategy can, at best, grow linearly in time, as opposed to exponentially for a fully cooperative strategy. However, tit-for-tat strategies are fairer than a fully cooperative strategy. Our results show that there exists a seed capacity threshold for tit-for-tat strategies. Increasing seed capacity beyond this threshold brings significantly reduced marginal gains.

1. Introduction

Peer-to-peer file sharing networks hold great promise as content distribution systems because of their ability to provide on-demand resource allocation. For example, P2P networks seem ideally suited to enable the online distribution of DVD quality movies at a fraction of the cost of a client/server downloading model. Smart cooperation among peers in a P2P file sharing network may enable the distribution of content to a large audience without the need to statically provision a large and expensive cyber-infrastructure ahead of time.

In our view, the cooperative nature of P2P networks is so important that among many possible characterizations of P2P networks within distributed systems, we choose individual peer behavior as our defining characteristic. In this context, distributed systems range from fully cooperative systems that employ a globally agreed upon pre-established strategy to completely noncooperative systems which, as we will show, represent the common client/server model. Most P2P networks fall somewhere in the middle of this range depending on their peer's selfish behavior.

There are many obstacles that P2P networks need to overcome to become the content distribution system of choice. One such obstacle is the free riding problem. Free riding peers use resources but do not contribute and, therefore, can hurt the network's overall service capacity. Many current P2P systems, notably BitTorrent [2], attempt to address the free riding problem by employing variations of tit-for-tat strategies [1]. However, the use of these strategies brings limitations to the performance of peer-to-peer networks as content distribution systems. These limitations have not received much attention in the literature.

In this paper, we provide an intuitive understanding of the limitations of P2P networks as content distribution systems when using tit-for-tat strategies. We obtain bounds on system performance and analyze a few optimal static peer strategies (and their corresponding outcomes) from two points of view:

- A global (macro) view providing overall system characteristics.
- An individual (micro) view providing the perspective of a single peer and what it is willing to do as it interacts with others in the network.

In short, we determine how each peer's unwillingness to contribute in tit-for-tat strategies limits the performance of P2P file sharing networks.

Our results also show that there exists a seed capacity threshold for tit-for-tat strategies. Increasing seed capacity beyond this threshold brings significantly reduced marginal gains.

The rest of the paper is organized as follows. After introducing the analysis framework, we describe the various cooperation models and then discuss the limitations of tit-for-tat strategies. Next, we investigate how increasing the seed capacity improves the expected download time. Finally, we discuss the literature on related models.

2. The Analysis Framework

To compare different scenarios, we use a very simple topological model where the only constraints are peer upload and download bandwidth. In this model there are no bottlenecks in the network core, all constraints are found at the edges. Peer i has a maximum uplink capacity of U_i bps and a maximum downlink capacity of D_i bps. We focus on uplink capacity because this enables us to clearly deal with selfish behavior, separating it from other concerns. In fact, we model an optimal infinite capacity network by only constraining each peer's upload bandwidth. In this case, the uplink capacity does not represent a physical network limitation but the peer's willingness to cooperate.

In this paper we always consider the worst case scenario of an *instant flash crowd*, where all N peers request a file of size Z bytes from the seed at time $t = 0$. The file is split into M pieces. We analyze P2P networks in four dimensions using the following measures:

- *Scalability*: Number of peers, N
- *Publishing workload*: Seed workload W and required upload capacity C
- *Efficiency*: Expected download time, $E[t]$
- *Fairness*: Normalized absolute peer workload imbalance, $\overline{I}_{Abs}$

A few remarks are appropriate here. We use the number of peers in the network and the seed workload/capacity as design parameters. The seed capacity is its uplink throughput capacity. The workload represents how many bytes of data it has to transfer to make sure all peers can obtain the content. A peer's download time is the time elapsed between when a peer joins the network to when it completes the desired download. Thus, the *expected* download time is of great interest to each peer (or the corresponding user) when it joins the network.

Measure of Fairness Peer i's imbalance is the difference in bytes between how much data the peer has downloaded d_i to how much it has uploaded u_i. Clearly, peers like positive imbalances and avoid negative ones. To calculate normalized imbalances, the normalization is done with respect to how much content has been distributed, *i.e.* the total amount downloaded by all peers. Thus,

$$\overline{I}_{Abs} \triangleq \frac{\sum_i |d_i - u_i|}{\sum_i d_i}$$

We do not consider the seed (*i.e.* the original publisher of the file) when performing the summations. Unlike the peers who want to download the content, the seed wants to distribute it. Excluding the seed from the summation leads to a metric that is more descriptive of peer behavior. We also assume peers will not download what they already have; *i.e.*, all downloaded content is new to the peer downloading it.

Using an alternative measure without the moduli would allow for two unfair transactions to cancel out. Similarly, using a maximum instead of a sum in the numerator or normalizing with respect to each peer instead of the whole system allows for unfair interactions to go undetected by the measure. We can also use a root-mean-square measure here, but we decided to stick to the absolute value measure for ease of interpretation.

The normalized imbalance is zero (cooperative) if all peers upload as much as they download, 1 ("completely noncooperative") if nobody but the seed uploads anything and 2 ("exploitative") if some peer is exploited and uploads to all others. More precisely, as the number of peers tends to infinity the normalized imbalance tends to zero in a perfectly fair network and to 2 in a perfectly unfair one. Values above 1 imply that at least one peer, who is not the seed, is altruistic/exploited and uploads more than it downloads. Furthermore, if no peer uploads more than it downloads then $\overline{I}_{Abs}$ measures what fraction of the file the peers did not reciprocate. Thus, if each client finishes its download and leaves the network after having uploaded on average only 30% of the file we have $\overline{I}_{Abs} = 0.7$. By this measure, being altruistic and uploading more than one downloads is not good and implies that the system is unfair. In this case, $\overline{I}_{Abs}$ can be greater than 1. For example, if we have a total of $N = 6$ peers (excluding the seed) and peer 1 uploads the whole file to the other 5 peers who free ride. Then,

$$\overline{I}_{Abs} = \frac{|1-5| + 5 \times |1-0|}{6} = 1.5$$

Again, the initial seed is not considered in these calculations. $\overline{I}_{Abs}$ can be computed for BitTorrent from the logs of the tracker if clients report their activity truthfully [3].

3. The Client/Server Model

In the client/server model, a server serving a file of size Z bytes has to do $W = ZN$ work to serve N peers. The workload increases linearly with the number of peers. If the server has upload capacity of C bytes/s then the earliest the first peer can finish getting the file is $\frac{Z}{C}$ seconds after the start, the second peer can only finish after another $\frac{Z}{C}$ seconds. Changing the order in which pieces are given to either peer can delay the time it takes for the first peer to finish, but it *cannot* shorten the total or average download time for both peers. It is easy to see by induction that the same is true for any number of peers.

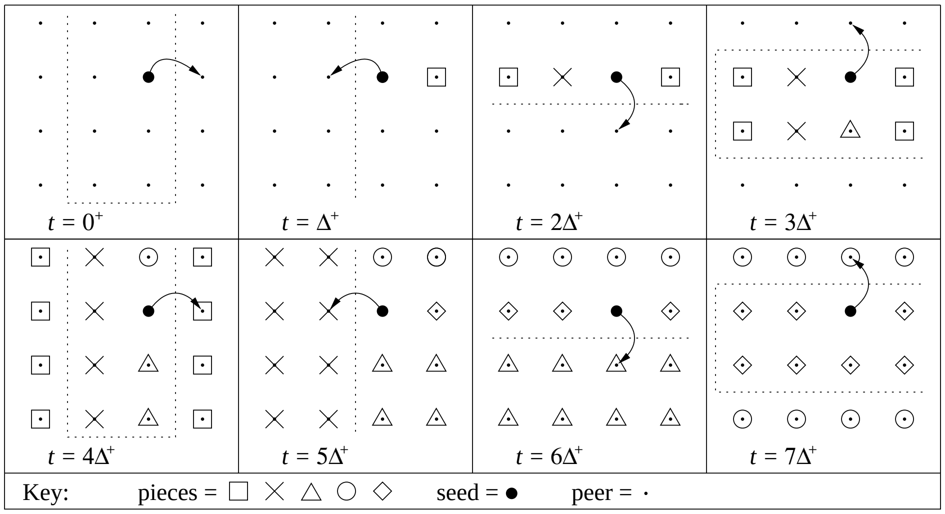

Figure 1. A Fully Cooperative strategy for $N = 16$ peers and $M = 5$ pieces. The figure shows what each peer has at the beginning of each time step. The arrows show the peer the seed is serving in the current time step. At the same time, peers upload to their "mirror images" across the dotted lines. Once all peers have a piece it is no longer shown (*e.g.* the square, $\Box$, disappears at $t = 5$). All peers complete the download in 8 time steps.

For an instant flash crowd of N peers, requesting the file at time $t = 0$, the expected download time is:

$$E\left[t\right] = \frac{1}{N}\left(\frac{Z}{C} + \frac{2Z}{C} + ... + \frac{NZ}{C}\right) = \frac{Z}{2C}(N+1)$$

In other words, the expected download time also increases linearly with the number of peers and is inversely proportional to the server capacity C. The client/server model is completely uncooperative as none of the clients contribute anything:

$$\overline{I}_{Abs} = \frac{\sum_i |d_i - u_i|}{\sum_i d_i} = \frac{\sum_i |d_i - 0|}{\sum_i d_i} = 1$$

4. A Fully Cooperative Strategy

Let us now perform the same analysis for a fully cooperative (FC) strategy proposed by [7]. The strategy is established ahead of time so that all peers know exactly what to do when the file becomes available. This simple strategy can be applied when $N = 2^k$ peers (including the seed), with all peers and the seed having equal upload capacity $C = U_i$. The file of size Z bytes is split into $M \geq k$ pieces of equal size which are uploaded to each peer in a pipeline as described below. Each peer can upload a piece as soon as it is completely downloaded. Because all pieces are of equal size we proceed as in discrete time steps of length $\Delta = \frac{Z}{MC}$. In each time step the seed gives out a new piece until the last new piece is given out. From that point on until all peers finish their downloads, the seed repeatedly uploads the last piece.

The strategy is described using Figure 1, where a file with $M = 5$ pieces, denoted by $\Box, \times, \triangle, \bigcirc$ and $\Diamond$, is distributed among $N = 16$ peers (including the seed). Pieces are distributed to peers as if each peer were the vertex of a 4-dimensional hypercube and four symmetric reflections were made successively. It should be clear that this strategy can be generalized for any $N = 2^k$. It has a number of nice features[1]:

- All peers finish the download simultaneously.
- Each peer only needs to connect to a small number ($\log_2 N$) of others.
- The download capacity required of each peer is the same as the upload capacity; *i.e.*, the strategy can be used when the download capacity is larger than or the same as the upload capacity of each peer.

The strategy is also very efficient. A peer cannot upload until it receives at least one piece. During each of the initial $k = \log_2 N$ steps every peer that has a piece continuously uploads to other peers. Therefore, the upload capacity of the network is used as much as possible. In each time step after the initial k, all peers but one (the seed's mirror image) completely use their upload capacity and one additional piece is

[1]Note that there are immediate extensions of this strategy to when the seed capacity is a power of two times the capacity of the other peers and when there are a total of 2^k peers *not* including the seed.

completed, *i.e.* all peers have a copy of it. The last two pieces are completed simultaneously. This implies that all peers finish their downloads at time step[2] $k+M-1$. Thus, the expected download time is given by:

$$E[t] = \Delta(k+M-1) = \tfrac{Z}{CM}(\log_2 N + M - 1)$$
$$= \tfrac{Z}{C}\left(\tfrac{M-1}{M} + \tfrac{\log_2 N}{M}\right)$$

In other words, for a fixed M the expected download time increases logarithmically with the number of peers.

The seed gives out a copy of each piece and then repeats the last piece for the very same $k+M-1$ steps. Thus, it gives out a total of:

$$W = \frac{Z}{M}(k+M-1) = Z\left(\frac{\log_2 N}{M} + \frac{M-1}{M}\right) \text{ bytes.}$$

Again, the increase in workload is logarithmic with an increase in the number of peers for a fixed M.

We can also calculate the normalized absolute imbalance (see appendix):

$$\overline{I}_{Abs} = \frac{N-2}{(N-1)M} + \frac{M-1}{(N-1)M} + \frac{2\log N}{(N-1)M} - \frac{(\log N)^2}{(N-1)M}$$

Notice that we cannot directly apply this strategy if $M \leq \log_2 N$. However, an immediate extension is to apply the strategy to a group of 2^M peers and then have peers recursively become seeds for other groups. For example, if $N = 1024$ and $M = 3$, we apply the strategy to a group of 8 peers and in 5 time steps we would have $8 = 2^3$ seeds. After another 5 time steps we would have a total of $2^3 \times 2^3$ seeds, and so on. For a fixed M, the resulting strategy still leads to an exponential growth in the number of cooperating peers and a respective logarithmic increase in the total expected download time as N increases.

In this section we included the seed in the number of peers N for ease of exposition. We will make the necessary change of variables when comparing this result with other strategies.

5. Tit-for-Tat Strategies

For a fixed number of peers, the fully cooperative strategy realizes large gains over the client/server model in terms of the expected download time and the seed workload, but it requires some peers to be altruistic and upload more than they download. Using the client/server and FC strategies as our benchmarks, we now examine two *non-exploitable* cooperative strategies in which peers only upload if they are simultaneously downloading. We will see that the slightly less cooperative behavior from the peers does limit the network's ability to scale, but we still achieve very advantageous results. We will consider two types of tit-for-tat strategies:

- **Direct Reciprocity:** Peer A uploads to peer B only if peer B is simultaneously uploading to peer A
- **Indirect Reciprocity:** Peer A uploads to peer B only if some peer uploads to peer A. This is more flexible and enables peers to form cycles, *e.g.* peer A uploads to peer B who uploads to C who uploads to A.

These restrictions do not apply to the seed who uploads at will, as always. The above restrictions imply that once a peer has all the pieces it stops cooperating. We assume for now that all peers and the seed have the same upload capacity.

Tit-for-tat strategies are quite fair. The peer behavior ensures that all such strategies have an imbalance $\overline{I}_{Abs}$ that is bounded by how much work the seed does. Assuming all peers employ the tit-for-tat strategy each peer must (directly or indirectly) receive its first piece from the seed to be able to participate. This implies that the seed must give out at least N pieces of the file. Similarly, for any strategy the seed must always give out at least one complete copy of the file (M pieces). Therefore, the workload $W = max(N, M)$ pieces or $W = \frac{Z}{M}\, max(N, M)$ bytes. Also, no peer will ever contribute more than it downloads and the terms inside the moduli in the expression for $\overline{I}_{Abs}$ are always positive. Therefore the imbalance is given by:

$$\overline{I}_{Abs} = \frac{\sum_i |d_i - u_i|}{\sum_i d_i} = \frac{\sum_i d_i - \sum_i u_i}{\sum_i d_i} = \frac{W}{NZ}$$

It follows that if $M \geq N$ the best possible imbalance for tit-for-tat strategies matches the best possible (smallest) imbalance for any strategy and is given by $\overline{I}_{Abs} = \frac{1}{N}$.

Because peers only cooperate after they get their first piece, in tit-for-tat strategies the number of cooperating peers only grows linearly with each time step as opposed to exponentially for the fully cooperative strategy. We will explore this in the next section.

A very efficient *indirect reciprocity* (IR) strategy can be obtained simply by forming a line where each piece is passed on to the next peer, just like the old fire brigade passed buckets to put out a fire. There are no restrictions on the number of peers. We proceed, as in the FC strategy, in time steps of length $\Delta = \frac{Z}{MC}$. Figure 2(a) illustrates the strategy for $N = M = 5$ peers. It should be clear that in the IR strategy of Figure 2(a) the peers also end the download in a sequence of steps. The first peer receives a constant download stream from the seed from $t = 0$ and finishes at time $t = \frac{Z}{C}$, the second finishes at $t = \frac{Z}{C} + \Delta = \frac{Z}{C} + \frac{Z}{MC}$

[2] If $M = 1$ the seed always serves the same piece and we are done in k timesteps.

and so on. Thus, the expected download time for the IR strategy is given by:

$$E[t] = \frac{1}{N}\sum_{i=0}^{N-1}\left(\frac{Z}{C} + i\frac{Z}{MC}\right) = \frac{Z}{C} + \frac{Z}{C}\frac{(N-1)}{2M}$$

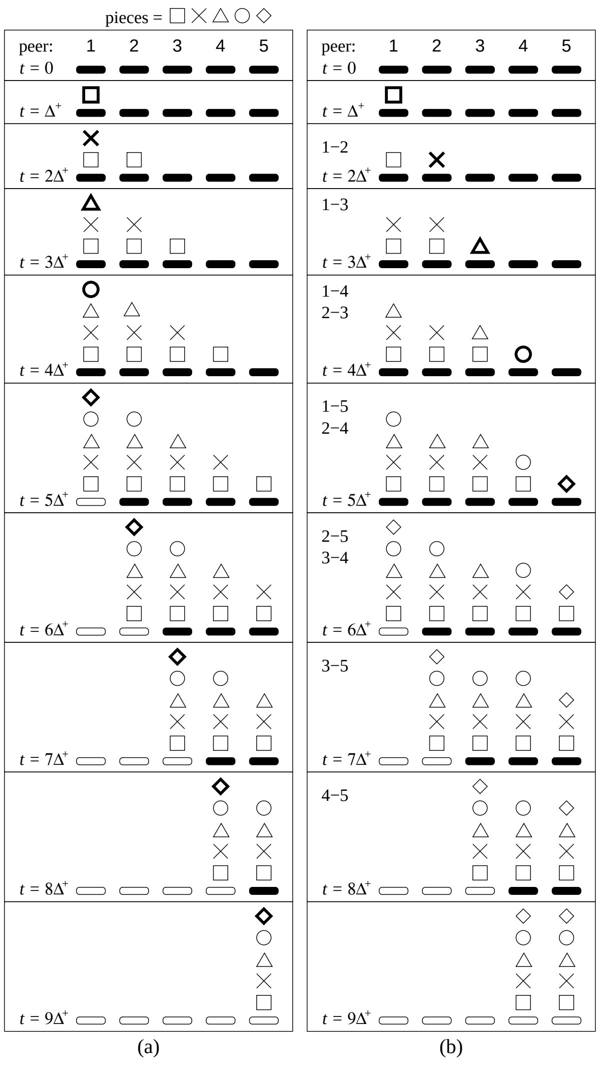

Figure 2. Two Tit-for-Tat Strategies using: (a) indirect reciprocity and (b) direct reciprocity. Peers in white have finished the download and no longer cooperate. Bold pieces have just been received from the seed in the previous time step. The numbers on the top-left corner of (b) show which peers are swapping pieces in the current time step. The left-most peer cooperates with the available rightmost peer as in a reflection across the middle.

Figure 2(b) shows a similar strategy that uses *direct reciprocity* (DR) only. It requires that the file be broken up into pieces such that $M = N$. From Figure 2, we can see that peers using the DR strategy finish one time step after their corresponding peers using the IR strategy, except for the last peer. It follows that the expected download time for this DR strategy is:

$$E[t] = \frac{1}{N}\left[-\Delta + \sum_{i=1}^{N}\left(\frac{Z}{C} + i\frac{Z}{MC}\right)\right]$$

$$= \frac{Z}{C}\left(\frac{3}{2} + \frac{1}{2M} - \frac{1}{M^2}\right)$$

where the $-\Delta$ term to the left of the sum is included because the last two peers finish simultaneously. It is very straight forward to calculate the workload W and the imbalance $\overline{I}_{Abs}$ for both strategies. These results are presented in Table 1 where we set $\delta = \frac{1}{2M} - \frac{1}{M^2}$.

The DR strategy requires that we partition the file so that there are as many pieces as there are peers. Unfortunately, for small files and large numbers of peers this many not be possible. Therefore we consider an immediate extension of this strategy to the case where $N = kM$ and $k \in \mathbb{N}$. The extended DR strategy separates the peers into cooperating groups of M peers and sequentially applies the DR strategy for $M = N$ to each group. Once the seed has done its work for one group, it immediately moves on to the next. We call this *sequential grouping*. Each group has M peers so that there are $k = N/M$ groups. Assume that the expected download time (measured from when the seed starts serving the group) for peers within the group is E_G. The seed is done serving a group after T_G seconds. Using sequential grouping the total expected download time for the extended DR strategy is given by:

$$E[t] = \frac{1}{k}\sum_{i=0}^{k-1}(E_G + iT_G) = E_G + \frac{T_G}{2}(k-1)$$

Substituting the appropriate values of k, T_G and E_G for the DR strategy we obtain:

$$E[t] = \frac{Z}{C}\left(\frac{3}{2} + \frac{1}{2M} - \frac{1}{M^2}\right) + \frac{Z}{C}\frac{(N-M)}{2M}$$

Notice that the seed workload for the *extended* DR strategy is still minimal among tit-for-tat strategies $W = kZ = NZ/M$. The total absolute imbalance is just the average imbalance for all groups. Thus, $\overline{I}_{Abs} = \frac{1}{M}$.

The same grouping idea can be used to extend the DR strategy to the case where $M > N$, more precisely $M = kN$ where $k \in \mathbb{N}$. We simply put all pieces in a pipeline and have the seed cycle k times through the N peers until it serves all M pieces of the file. The peers also repeat their actions k times. This pipelined DR strategy also has minimal workload.

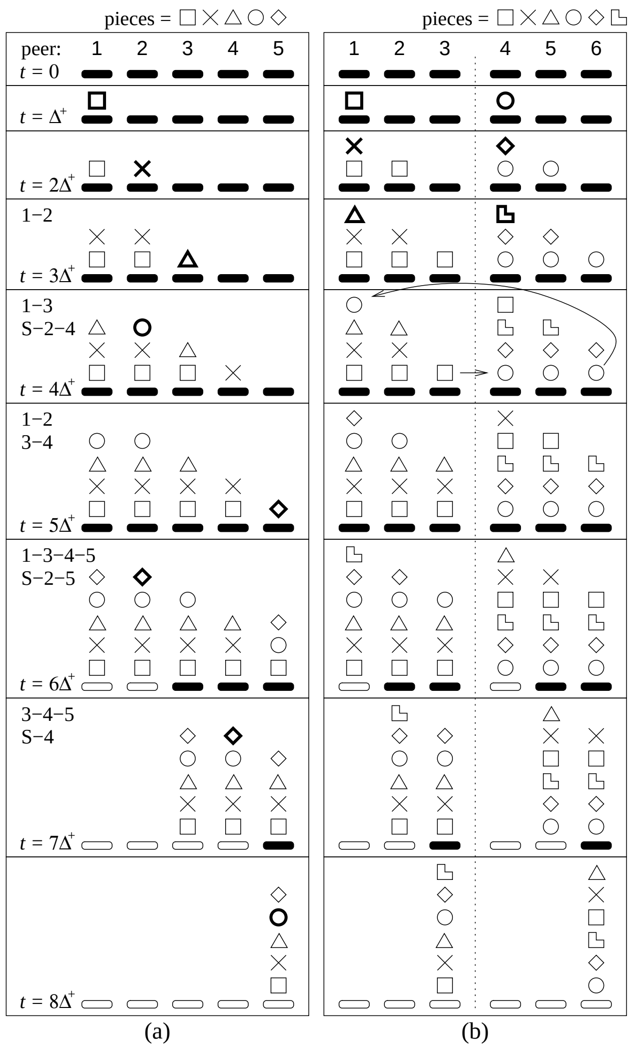

Figure 3. (a) Fast IR Strategy that can be used when there are no download constraints. Some peers receive more than one piece in a single time step. (b) Parallel Grouping for $M = N = 6$ and $s = 2$.

6. Limitations of Tit-for-tat Strategies

The reason for considering these two non-exploitable strategies is their optimality. In the IR strategy, every participating peer except for the last one uploads as much as possible, as soon as possible, for as long as possible for any tit-for-tat strategy. Thus, for an infinite N this is the fastest possible tit-for-tat strategy[3]. This is also the fastest possible strategy for finite N if we have download constraints $D_i = U_i = U = C$. If we have download constraints, the pigeonhole principle can be iteratively applied to show that the last peer cannot upload without violating the download constraints. However, if we do not have download constraints and N is finite there are faster strategies. Figure 3(a) shows a counterexample.

[3]In other words, for a sufficiently large l, the average download time of the first l peers is no greater than that of any other strategy.

In any strategy, the seed must give out at least one complete copy of the file. This is exactly the workload of the DR strategy. Thus, the DR strategy for $N = M$ matches the smallest seed workload of any strategy. Similarly, it should be clear from the discussion in the beginning of section 5 that the DR strategy matches the best (smallest) imbalance of any strategy.

Table 1 presents our exact results. Please take a while to examine it. In the table, the similarities between the results for the IR strategy and the client/server model are striking! For these two strategies, both W and $E[t]$ differ only by the inclusion of the parameter M. The table also shows that it is always beneficial to increase M, it leads to smaller workloads and faster download times for all strategies.

Peers in tit-for-tat strategies can only cooperate with at most $M - 1$ other peers before finishing the download because after a peer receives one piece and is able to cooperate, there are only $M - 1$ other pieces the peer wants to download. This bound on the number of cooperating peers means that tit-for-tat strategies cannot scale as well as fully cooperative strategies. Furthermore, the parameter M determines exactly what gain in upload capacity the IR strategy will have over the client/server model once as many peers as possible are cooperating. This can be quite significant. For example, a typical use of BitTorrent to download a 1GB movie file with the standard piece size of 256KB implies $M = 4096$ pieces. However, for even larger N the FC strategy is the clear winner.

It is interesting to note that, neglecting the first piece received by each peer in the DR strategy, in both strategies the peers get the pieces in the same order that the seed gives them out. This would be useful for streaming applications.

We can extend these strategies to a scenario where peers have different upload capacities by establishing cooperating groups. To do so, we let a peer select a different cooperating group per kbps of upload rate it has. For example, if 5 peers have 75kbps of upload bandwidth and another 100 peers only have 50 kbps. We can form two cooperating groups, one including all peers and uploading at 50 kbps and another formed only by the high speed peers, that upload between themselves at an extra 25 kbps.

7. Seed Capacity

The results in Table 1 can be misleading and appear to suggest that a linear increase in seed capacity C would cause a similarly linear decrease in $E[t]$ for all strategies. This is not true. The results for all the cooperative strate-

Measure	Fully Cooperative	Indirect	Extended Direct	Client/Server
W	$Z + Z\frac{\log_2(N+1)-1}{M}$	$Z + Z\frac{N-1}{M}$	$Z + Z\frac{N-M}{M}$	$Z + Z(N-1)$
$E\left[t\right]$	$\frac{Z}{C} + \frac{Z}{C}\frac{\log_2(N+1)-1}{M}$	$\frac{Z}{C} + \frac{Z}{C}\frac{N-1}{2M}$	$\left(\frac{3}{2}+\delta\right)\frac{Z}{C} + \frac{Z}{C}\frac{N-M}{2M}$	$\frac{Z}{C} + \frac{Z}{C}\frac{N-1}{2}$
$\overline{I}_{Abs}$	$\frac{N-1}{NM} + \frac{M-1}{NM} + \frac{2\log_2(N+1)}{NM} - \frac{(\log_2(N+1))^2}{NM}$	$\frac{1}{N} + \frac{1}{M} - \frac{1}{NM}$	$\frac{1}{M}$	1

Table 1. Exact Workload, Expected Download Time and Normalized Imbalance for different Strategies

gies in Table 1 assume that each peer's upload capacity is the same as the seed's, *i.e.* $C = U_i = U \ \forall i$. In fact, a linear increase in capacity, that is not matched by a similar increase in peer capacity U in the FC strategy, can only cause a logarithmic decrease in $E[t]$ (see parallel grouping below). On the other hand, the client/server model always has the desired reciprocal relationship between C and $E[t]$. This leads us to consider what gains can be obtained by increasing the seed capacity in tit-for-tat strategies.

We now extend these strategies to when the seed capacity is a multiple of the peer capacity; *i.e.*, $C = sU$ and $s \in \mathbb{N}$. In other words, the seed can upload s pieces per time step as opposed to a peer who can upload only one piece. This is equivalent to either increasing the bandwidth or replicating the seed. We will assume that the seed splits its total upload capacity into s different "pipes" where each individual pipe will have the same capacity as a peer. It is easy to see that this strategy is not optimal for $E[t]$ by considering the client/server model with only two clients. If instead of equally spliting its upload capacity between the two clients a server uploads at full capacity to each one, in turn, the client's expected download time decreases. However, the decrease for P2P networks is not so significant because the file is split into pieces and peers cooperate among themselves. The first peer's first piece can be downloaded in Δ/s seconds from a seed with s times the upload capacity of a peer; but even if we establish this as a lower bound on the time it takes to upload the first piece to *all* peers, we can see that this lower bound implies that this strategy can decrease the total expected download time by at most Δ.

Consider the case where $N \gg M$. Again, we split the seed's capacity into s different "pipes" where each individual pipe will have the same capacity as a peer. We can then employ the tit-for-tat strategies just described using each pipe as a seed that has the same upload capacity as a peer. This will make s distinct groups of cooperating peers work in parallel. We call it *parallel grouping*. Using this extension with the IR strategy, we can serve s groups of M peers in the same time it takes to serve just one group of M peers with a low capacity seed. Therefore, the ratio of expected completion times is given by:

$$\frac{E[t|\,high\,capacity]}{E[t|\,low\,capacity]} = \frac{N + (2M-1)}{sN + (2M-1)}$$

For $N \gg M$, increasing the seed capacity by a factor of s decreases the total expected download time by a factor of s.

We can easily apply parallel grouping as long as $s \geq N/M$. In this case, each group will have at least M peers and exactly M pieces. However, if we continue to increase the value of the seed capacity C, parallel grouping cannot maintain a corresponding linear decrease in the expected total download time.

For large values of s, there are more pieces of the file than there are peers in each cooperating group, *i.e.*, $M > \frac{N}{s}$. We can use this, coupled with the large seed capacity, to further decrease the expected download time. In fact, for cooperating groups of $G = \frac{N}{s}$ peers each, where the number of pieces is a multiple of the number of cooperating peers in the group, we can get very close to the optimal (fastest) tit-for-tat strategy with the strategy illustrated in Figure 3(b) for $M = 6$ pieces, $N = 6$ peers and $s = 2$. In Figure 3(b), there are 2 groups of cooperating peers. The seed is able to give different pieces to each group because there are twice as many pieces as there are peers in each group. This, in turn, enables the groups to cooperate to finish the download. In Figure 3(b), the seed only gives out a single copy of the file to the two groups and then stops. This is the minimum workload. The strategy presented in Figure 3(b) is optimal for the minimal workload. However, if the seed continues to work and carefully chooses who to serve until all peers finished their downloads we then obtain the fastest tit-for-tat strategy given the parameters N, M and s.

Due to space considerations we will only present this strategy for the special case where $s = N$. In other words, the seed capacity is the same as that of all peers put together, $C = NU$. This represents an M fold increase in seed capacity over case where $s = N/M$. However, despite the huge increase in capacity, we now show that the optimal tit-for-tat strategy only provides a small decrease in the expected download time.

The optimal strategy for $s = N$ is very simple. In the first time step, the seed simply uploads different pieces to all the peers simultaneously. After each peer receives its first piece it joins a large uploading cycle that includes all N peers. There, in each time step, it uploads the last piece it received. During each time step peers simultaneously receive a piece from the seed. Because each peer is receiving two pieces per time step (except for the first piece) all peers finish the download in $1 + \lceil \frac{M-1}{2} \rceil$ time steps and we have that the expected download time is:

$$E_{s=N}[t] = \Delta \left(1 + \left\lceil \frac{M-1}{2} \right\rceil \right)$$

If we compare this number with the expected download time when $s = N/M$, we obtain:

$$\frac{E_{s=N}[t]}{E_{s=\frac{N}{M}}[t]} = \frac{\Delta \left(\frac{2}{2} + \lceil \frac{M-1}{2} \rceil \right)}{\Delta \frac{3M-1}{2}} \approx \frac{M+1}{3M-1} > \frac{1}{3}$$

The above result shows that, unlike what happens in the client/server model, in a tit-for-tat strategy there are decreasing gains as one increases the seed upload capacity. This occurs because once the seed capacity s reaches the ratio N/M, the optimal P2P strategy starts to transition into a client/server model and the seed takes more and more of the workload that was being provided by the cooperating peers. Thus, we suggest the following rule-of-thumb to determine the optimal seed capacity for P2P systems employing tit-for-tat strategies: $C^* = \frac{N}{M}U$.

8. Related Work

The work of Yang and de Veciana [7] is likely the closest to this one. That work analyzes the service capacity of P2P networks under transient and steady-state regimes. The deterministic model used in this work is very similar to the simpler (non-branching) detereministic model used by Yang and de Veciana for transient regimes. However, Yang and de Veciana go on to consider a more complicated branching model that focuses on other characteristics of real P2P networks, such as peer churn. In their transient analysis, each peer decides whether to leave the system (with probability ζ) or to continue to upload (with probability $1 - \zeta$) after completing the download. Our work focuses on tit-for-tat strategies, where peer behavior is not so altruistic, and the limitations these strategies bring.

Qiu and Srikant [6] analyze BitTorrent like systems with a very interesting fluid model. Although they do consider peer strategy in detail, their analysis focuses on steady-state performance and the existence of Nash equilibrium strategies.

The very general queueing network model of Ge *et al.* [4] can be applied to a variety of P2P networks and can used as a means for comparison. They focus on presenting a framework that unifies three different charactersitic of P2P networks: infrastructure maintenance, query handling and system throughput. However, their definition of "free-loaders" is not strong enough to give a precise characterization of the effects peer behavior on performance.

9. Conclusion

We believe a content publisher chooses which peer-to-peer content distribution systems to use based on 3 parameters: number of peers, publishing workload and expected download time. From this point-of-view, fairness is important only as much as it allows for better overall service. The client/server model is very inefficient both in terms of workload and download time (when compared to the other cooperative strategies), especially as the number of peers increases. It requires more work and it takes longer to upload/download files.

Because of the optimality of the fully cooperative strategy, it should be clear that, for a fixed number of pieces M, the expected download time for any cooperative content distribution system cannot do better than increase logarithmically with the number of peers N. The parameter M is very important for tit-for-tat strategy. It determines the overall capacity of the system by establishing how many peers can simultaneously cooperate.

In the ideal topological model considered, there is no significant difference between the IR and DR strategies. However, the connectivity requirements for each are different. The DR strategy requires that each peer connect to another $M-1$ peers, whereas in the IR strategy each peer only connects to two others (in all tit-for-tat strategies the seed must connect with all peers). We conjecture that other topological models will show differences in performance between the two strategies.

We also observe that, for systems employing tit-for-tat strategies to deal with flash crowds, there is an efficiency threshold that when crossed reduces the marginal gains obtained by increasing the seed capacity.

References

[1] R. Axelrod, *The Evolution of Cooperation*, Basic Books, New York, 1984.

[2] B. Cohen, "Incentives build robustness in BitTorrent". In *Workshop on Economics of Peer-to-Peer Systems*, Berkeley, CA, USA, June 2003. `http://bittorrent.org`

[3] BitTorrent Protocol Specs. `http://bittorrent.org/protocol.html`

[4] Z. Ge, D. R. Figueiredo, S. Jaiswal, J. Kurose and D. Towsley, "Modeling Peer-Peer File Sharing Systems". In *Proceedings of IEEE INFOCOM*, 2003.

[5] Gnutella `http://www.the-gdf.org`

[6] D. Qiu and R. Srikant, “Modeling and Performance Analysis of BitTorrent-Like Peer-to-Peer Networks”. In *Proceedings of ACM SIGCOMM*, 2004.

[7] X. Yang and G. de Veciana, “Service Capacity of Peer to Peer Networks”. In *Proceedings of IEEE INFOCOM*, 2004.

Appendix

Calculating the Imbalance for the FC Strategy

In the FC strategy, peers start uploading as soon as they get their first piece and all peers finish the download simultaneously. The seed uploads to the first peer in the first time step. In the second time step, the seed and the first peer upload to two other peers (along a different line of symmetry). *Including the seed as a peer,* 4 peers recieve a piece at the end of the second time step. Similarly, 8 peers recieve their first piece at the end of the third time step and so on. All $N = 2^k$ peers will recieve their first piece at the end of k steps.

The FC strategy lasts for a total of $k + M - 1$ time steps. Let us split the analysis into the first k time steps when some peers are idle and the remaining $M - 1$ steps when all the peers are uploading. In the k^{th} step the last $N/2$ peers recieve their first piece; thus, they upload 0 pieces in the first k steps. In the $(k-1)^{th}$ step, $N/4$ peers receive their first piece. These $N/4$ peers upload one piece in the next time step. Similarly, $N/8$ peers upload 2 pieces in the first k steps and so on.

After the first k time steps, all the peers have a piece and will upload in each of the remaining $M - 1$ steps. However the peers which are “mirror images” of the seed do *not* upload when they are downloading from the seed. For each of these peers this occurs once in every $\log N$ steps. Defining $B = M - 1$, such that B is the “basal” workload of all peers, the above argument can be summarised by saying that during the entire download:

- $\frac{N}{2} - 1$ peers upload $0 + B$ pieces
- $\frac{N}{4} - 1$ peers upload $1 + B$ pieces
- $\frac{N}{8} - 1$ peers upload $2 + B$ pieces
- $\frac{N}{2^i} - 1$ peers upload $(i-1) + B$ pieces

The reason we subtracted one peer from each of the above sets is that these are “the mirror image peers” and they upload $\left(\frac{B}{\log N}\right)$ less than their respective groups. For example, the mirror image peer for the second group above uploads $1 + B - \left(\frac{B}{\log N}\right)$ pieces. Note that, as required, the seed workload is not counted here.

All peers download the complete file of M pieces. Therefore,

$$\bar{I}_{Abs} = \frac{\sum_i |d_i - u_i|}{\sum_i d_i} = \frac{\sum_{i=1}^{N-1} |M - u_i|}{\sum_{i=1}^{N-1} M} = \frac{1}{(N-1)M} \sum_{i=1}^{N-1} |M - u_i|$$

Where the seed is excluded from the summations. Expanding the last sum we obtain:

$$\left[\left(\frac{N}{2}-1\right)|M-B| + \left(\frac{N}{4}-1\right)|M-(B+1)| + \left(\frac{N}{8}-1\right)|M-(B+2)| + \dots + \left(\frac{N}{N}-1\right)|M-(B+\log N-1)|\right] + \beta$$

$$= \left[\frac{N}{2}|M-B| + \frac{N}{4}|M-B-1| + \frac{N}{8}|M-B-2| + \dots + \frac{N}{N}|M-B-(\log N-1)|\right] - \alpha + \beta$$

where:

$$\alpha = \sum_{i=1}^{k} |M - (i-1+B)| = 1 + \frac{(k-2)(k-1)}{2}$$

$$\beta = \sum_{i=1}^{k} \left| M - \left(i - 1 + B - \frac{B}{\log N}\right)\right| = M - 1 + \frac{1}{2}\left(3k - k^2\right)$$

and the last equality for β holds when $\log N \leq 1 + \sqrt{M}$. Therefore,

$$\bar{I}_{Abs} = \frac{1}{(N-1)M}\left[\frac{N}{2}\times 1 + \frac{N}{4}|1-1| + \frac{N}{8}|1-2| + \dots + \frac{N}{N}|1-(\log N-1)|\right] + \frac{\beta-\alpha}{(N-1)M}$$

$$= \frac{1}{(N-1)M}\left[\frac{N}{2} + \left(\frac{N}{8}(1) + \frac{N}{16}(2) + \frac{N}{32}(3) \dots + \frac{N}{N}(\log N - 2)\right)\right] + \frac{\beta-\alpha}{(N-1)M}$$

We recall that:

$$\frac{1}{2} + \frac{2}{4} + \frac{3}{8} + \frac{4}{16} + \dots + \frac{\log N}{N} = 2 - \frac{2+\log N}{N}$$

Finally, using this result and simplifying we obtain:

$$\bar{I}_{Abs} = \frac{N-2}{(N-1)M} + \frac{M-1}{(N-1)M} + \frac{2\log N}{(N-1)M} - \frac{(\log N)^2}{(N-1)M}$$

The four terms above can be interpreted as follows**.** The first term stems from the fact that each peer receives a piece before it begins to upload and stops uploading immediately once it has all pieces. Therefore, each peer receives at least one more piece than it uploads among the M pieces of the file. The second term comes from the fact that once all peers have a piece, at each step, one in $N - 1$ peers does not upload. Finally, the last terms are due to the unreciprocated work that is done before all peers have a piece.