Gaining Intuition through a Peer-to-Peer Comparison Framework

Dimitri do B. DeFigueiredo, Balaji Venkatachalam and S. Felix Wu

Department of Computer Science – UCDavis

Abstract

We propose an analysis framework for peer-to-peer (P2P) networks and use it to compare a few optimal content distribution strategies. We propose new strategies and obtain bounds on the performance of any P2P network depending on individual peer behavior. The analysis provides insights on how P2P networks perform when dealing with flash crowds.

1 Introduction

Is BitTorrent [1] better than Gnutella [3]? How do we compare different peer-to-peer (P2P) filesharing networks? In this paper we propose an analysis framework for P2P networks and use it to compare a few optimal content distribution strategies. Among the 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 an overall centralized 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 selfish behavior.

To provide an intuitive understanding of our framework we analyze a few idealized static peer strategies and their corresponding outcomes. We consider the 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.

Here we focus on *content distribution* through filesharing networks, but the reasoning holds for any *service network*. In that case, quality of service, rather than bandwidth, is the limiting factor.

2 The Framework

We use a very simple topological model where the 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.

The proposed framework analyzes P2P networks in four dimensions: *scalability, publishing workload, efficiency and fairness*. We choose to quantify these respectively by the following measures:

- Number of peers, N
- Seed workload W and required upload capacity C
- Expected download time, $E[t]$
- Normalized absolute 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 MBytes 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. 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[1]. 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. 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 $\overline{I}_{Abs}$

[1] We can also use a root-mean-square measure here. We decided to stick to the absolute value measure for ease of interpretation.

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 [2].

3 The Client/Server Model

In the client/server model, a server serving a file of size Z MB 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 MB/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.

In this paper we always consider the worst case scenario of an *instant flash crowd*, where all N peers request the file at time $t = 0$. So, 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 [4]. 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. The file of size Z MB is split into M 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 $\square, \times, \triangle, \bigcirc$ 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[2]:

- All peers finish the download simultaneously.
- Each peer only needs to connect to a small number ($\log N$) 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 at least one 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 completely use their upload capacity and one additional piece is 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[3] $k+M-1$. Thus, the expected download time is given by:

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

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):

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

[2] 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.

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

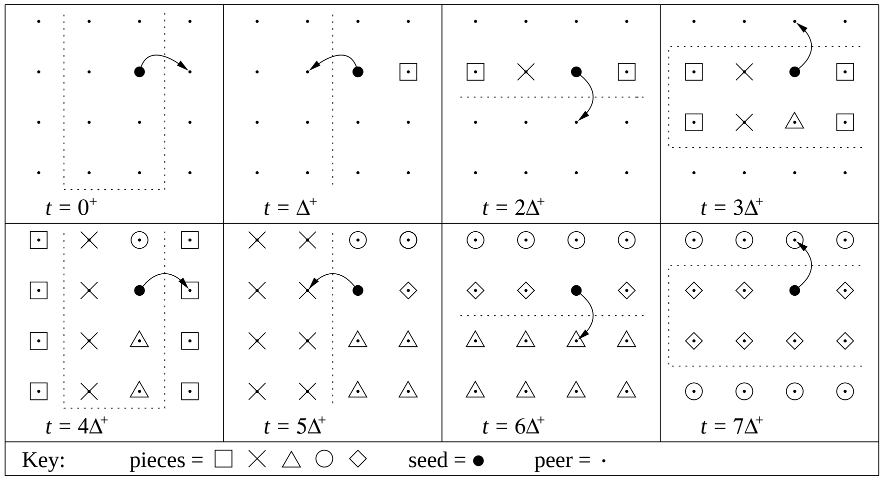

Figure 1: A Fully Cooperative strategy for $N = 16$ peers and $M = 5$ pieces. The figure shows what each peer has at the beginning of each time step. The arrows show the peer the seed is serving in the current timestep. At the same time, peers upload to their "mirror images" across the dotted lines.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

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. We now examine two *non-exploitable* cooperative strategies in which peers only upload if they are simultaneously downloading. We will see that despite the slightly less cooperative behavior from the peers we still achieve very advantageous results. We will consider two types of tit-for-tat strategies:

- **Direct Reciprocity:** Peer A uploads to peer B iff peer B is simultaneously uploading to peer A
- **Indirect Reciprocity:** Peer A uploads to peer B iff 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 that all peers and the seed have the same upload capacity[4].

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 receive at least one piece from the seed to be able to participate. 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:

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

Note that the seed must always give out at least a complete copy of the file, so $W \geq Z$ and it follows that the best possible imbalance for tit-for-tat strategies matches the best possible imbalance for any strategy and is given by $\overline{I}_{Abs} = \frac{1}{N}$.

The fact that in tit-for-tat strategies each peer must receive at least one piece from the seed to be able to participate implies that the seed must upload to each one of N peers. Also, because peers only cooperate after they get their first piece, the number of peers cooperating only grows linearly with each time step as compared to exponentially for the fully cooperative strategy.

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, but we require that the file is broken up into pieces such that $M \geq N$. We proceed, as in the FC strategy, in time steps of length $\Delta = \frac{Z}{MC} = \frac{Z}{NC}$. Figure 2(a) illustrates the strategy for $N = M = 5$ peers. Figure 2(b) shows a similar strategy that uses *direct reciprocity* (DR) only. It should be clear that in the IR strategy, 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

[4] There are immediate extensions of these strategies when the capacity of the seed is a multiple of the capacity of the peers. We simply form groups of peers and apply the same strategy in each group.

second finishes at $t = \frac{Z}{C} + \frac{Z}{NC}$ and so on. Thus, the average download time for the IR strategy is given by:

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

Similarly, for the direct reciprocity strategy we have:

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

Where the negative 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 the next section.

The reason for considering these two non-exploitable strategies in particular is their optimality. In the IR strategy, every participating peer uploads as much as possible, as soon as possible, for as long as possible for any tit-for-tat strategy. Thus, this strategy matches the smallest possible expected download time of any tit-for-tat 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 matches the smallest seed workload of any strategy. Similarly, it should be clear from the discussion above that the Direct Reciprocity strategy matches the best (smallest) imbalance of any strategy.

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, which would be interesting for streaming applications.

6 Discussion

Table 1 presents our exact results. For comparison with the client/server model the seed is excluded and does *not* count as one of the peers. This requires us to make a change of variables to the results obtained for the fully cooperative strategy to align it with the others.

To simplify the comparison, we eliminate M by assuming a fixed piece size of 1 unit (*i.e.* 1MB) in the FC strategy, this implies that $M = Z$. The client/server model is quite effective when the number of clients is small. Thus, we are mostly interested in analyzing peer-to-peer systems as a content distribution platform that can cope with large numbers of peers in flash crowds. Note also that use of BitTorrent to download a 1GB file with the old piece size of 1MB implies $M = 1000$ pieces. Thus, we assume M and N to be large and make the approximations: $\log_2(N+1) \approx \log_2 N$, $\frac{N-1}{N} \approx 1$ and $\frac{M-1}{M} \approx 1$. With these approximations we obtain the results in Table 2. Please take a while to examine it.

Interestingly, the expected download time for the DR and IR strategies is independent of the number of peers for large N. This needs to be taken with a grain of salt and is a consequence of having to choose a large M for these strategies to work. In the DR and IR strategies we required that[5] $M \geq N$. If we choose $M = N$ for the FC strategy the expected download time actually decreases as N increases tending to the limit $\frac{Z}{C}$ (which is the expected time for just 1 peer).

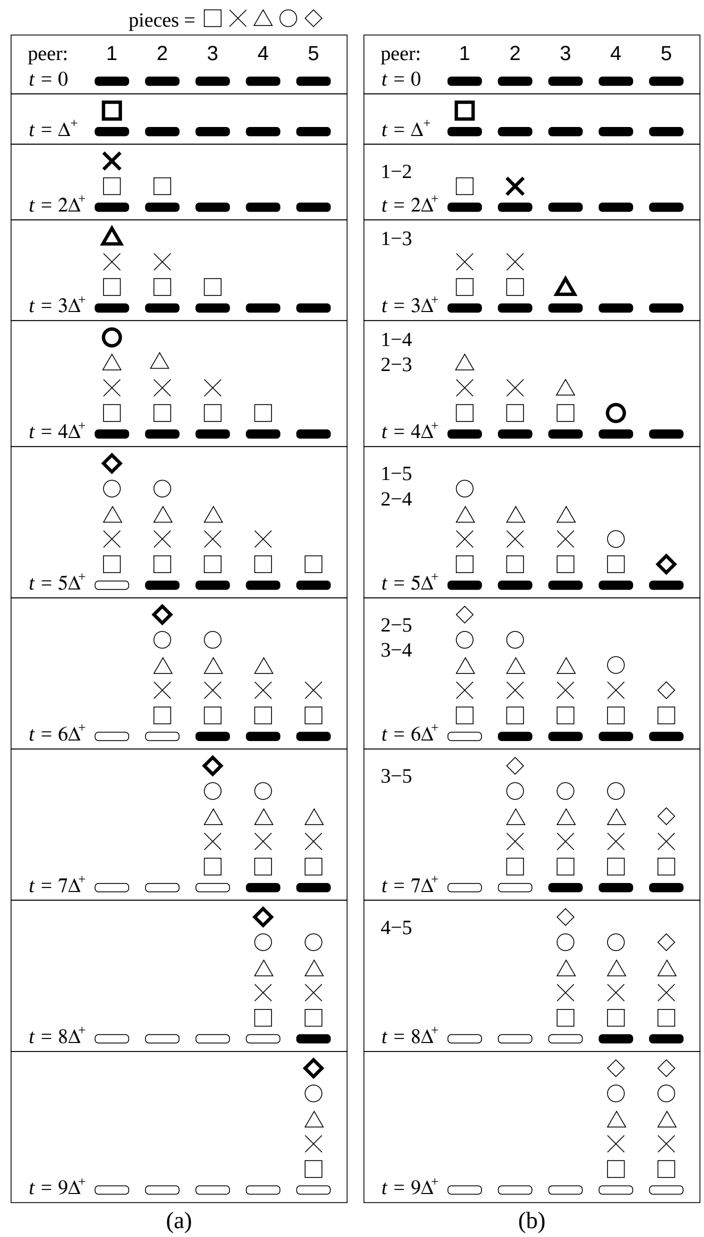

Figure 2: Two Tit-for-Tat Strategies using: (a) indirect reciprocity and (b) direct reciprocity. Peers in white have finished the download and no longer cooperate. Bold pieces have just been received from the seed in the previous time step. The numbers on the top-left corner of (b) show which peers are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

[5]We can also pipeline the DR strategy if $M = kN$ for $k \in \mathbb{N}$.

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

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

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

Table 2: Summary of Results

We believe a content publisher would choose which peer-to-peer content distribution systems to use based on 3 axes: number of peers, workload and expected download time. From this point-of-view, fairness is important only as much as it allows for better overall service. It should be clear that the client/server model is very inefficient both in terms of workload and download time (when compared to the other strategies) as the number of peers increases. It requires more work and it takes longer to upload/download files. We can calculate what the gains in workload and capacity from using a cooperative strategy are. We define these gains as the ratio of the workload W and the capacity C needed to serve the same number of peers with the same expected download time in comparison to the client/server model.

Because of the optimality of the fully cooperative strategy, it should be clear that the expected download time for any cooperative content distribution system cannot do better than increase logarithmically with the number of peers N.

One interesting difference between the FC and the non-exploitable tit-for-tat (IR and DR) strategies is that FC allows for the number of cooperating peers to increase exponentially with time; whereas, the non-exploitable strategies only allow for a linear increase. This limitation stems from the fact that in the non-exploitable strategies the seed must give out at least one piece to each peer before it can cooperate with others. In practical terms, this may limit how fast tit-for-tat strategies scale to handle large flash crowds when file size is small. In the very idealized scenario considered this is the only significant loss we have from adopting non-exploitable strategies rather than the fully cooperative one. Further investigation is necessary to determine the extent to which this may be mirrored in the real world. However, the gains of non-exploitable strategies over the client/server model are still very significant.

Although it may be tempting to compare the expected download times of the IR and DR strategies, such comparison would only be appropriate if the strategies had the same workload. The static scenarios considered are unable to show any significant difference between direct and indirect reciprocity with both achieving close to optimal performance. We conjecture that a more realistic setting where there is significant peer turn over may be able to differentiate between the two.

The seed capacity was inversely proportional to the expected download time for all the strategies considered. It appears that a linear increase in capacity can be effectively used to produce a corresponding linear decrease in expected download time. We can relax the assumptions made on peer upload capacity by establishing extended strategies where a peer selects different cooperating groups 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[6]. However, we made very strong assumptions about how much each peer can upload compared to the seed. Thus, this question requires further investigation.

Acknowledgments

The authors would like to thank Earl Barr for many insightful conversations.

References

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

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

[3] http://www.the-gdf.org

[4] X. Yang and G. de Veciana. "Service Capacity of Peer to Peer Networks" In INFOCOM, 2004.

[6]Note that the seed can *force* cooperation between peers by determining where peers should get their next pieces from.

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,

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

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

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

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

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,

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

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

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:

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

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.