# Minimising Peer On-Time for Energy Efficient Peer-to-Peer File Distribution

Andrew Sucevic, Lachlan L. H. Andrew, Thuy T. T. Nguyen
Centre for Advanced Internet Architectures, Technical Report 120809A
Swinburne University of Technology
Melbourne, Australia
andrew_sucevic@iprimus.com.au, landrew@swin.edu.au, tnguyen@swin.edu.au

***Abstract*—Peer to peer (P2P) techniques are an effective means of content distribution. By exploiting the upload bandwidth of peers, P2P can reduce the time required for file distribution, yet requires the peers to stay active to participate in the distribution process (for downloading, uploading or both). So far little attention has been paid to its energy consumption. This paper studies lower bounds on the possible total time that peers need to be active in order to distribute a file from a single server to a given set of peers. We show that a P2P system optimized for energy efficiency can consume half the energy of one optimized purely to minimize download time, while imposing minimal additional delay. To achieve this, peers should finish in increasing order of upload capacity, which is the reverse of what is optimal for delay. Moreover, peers should turn off as soon as they finish downloading. For networks of up to three peers, optimal strategies are derived and proven to be optimal, and larger systems are studied by simulation.**

## I. Introduction

Personal computers and their monitors have been estimated to consume a quarter of all ICT energy [1]. A large fraction of this is due to corporate policy which requires that PCs be left on overnight to allow administration such as software upgrades. This motivates the study of energy-efficient file distribution schemes.

Peer to peer (P2P) technology can reduce the time required for file distribution, but little attention [2], [3] has so far been paid to its energy consumption.

This paper considers lower bounds on the possible total time that peers need to be active, either uploading or downloading, in order to distribute a file from a single server to a given set of peers. Such bounds are useful both because they give a benchmark against which to compare implementable schemes, and also because they can give structural insights such as indicating the optimal order in which peers should complete.

Following [4]–[6], only upload constraints are considered, while network congestion and download constraints are ignored. It is assumed that a set of peers simultaneously start downloading a given file, and that as soon as a peer has downloaded some data, that data can be forwarded to other peers.

In this model, a simple symmetric strategy minimizes the time for the last peer to receive the file [4], [7]. This strategy causes all peers to finish simultaneously. However, some peers can finish much earlier with minimal or no increase in the finish time of the last peer. This led to a search for smaller average finish times [8]–[11]. It was conjectured in [11] that sequentially minimizing finish times also minimizes the sum of finish times, but a counterexample was presented in [12]. For a given finishing order, [6] derives the polytope of all possible combinations of finish times.

However, these schemes assume that peers will remain on to continue uploading until the last peer has finished downloading. This need not minimize the energy consumed by the peers. This paper instead considers the case that peers may stop uploading before the last peer has finished, to enable them to be turned off. It is shown that it is often optimal to turn peers off as soon as they have finished downloading, which has the side benefit of not requiring finished peers to continue uploading out of altruism.

A mathematical model of this system is presented in Section II. For networks of up to three peers, strategies which minimize energy consumption are derived in Section III and proven to be optimal. Notably, in these cases peers finish in a different order from that which is optimal when only delay is considered, and peers should turn off as soon as they finish downloading. Larger cases are studied by simulation in Section IV, where it is shown that a simple strategy can halve the energy consumption relative to the schemes of [4], [7].

## II. Model

The system considered here has a single server which contains a file of size $F$ which is distributed to $N$ peers. All nodes (server and peers) are assumed to be able to communicate to all other peers, with the only constraint being the upload capacity of each node. The network is also assumed to be static, in that no peers can arrive or leave. The file is broken up into infinitely small pieces, which allows a peer to immediately forward any data it receives without delay to another peer.

The following notation is used in this model:

- $F$: size of the file
- $F_i(t)$: amount of file that peer $i$ has at time $t$
- $N$: total number of peers (not including the server)
- $\tau_i$: *download time* of the $i$th peer to finish; $\tau_i \leq \tau_{i+1}$
- $t_i$: *turn off time* of the $i$th peer to finish; $t_i \geq \tau_i$
- $C_s$: upload capacity of the server
- $C_j$: upload capacity of peer $j$. Without loss of generality these are in decreasing order: $C_j \geq C_k$ given $j < k$
- $p_i$: the $i$th peer to finish. When written as a subscript, this is written $pi$; for example, $C_{p1}$ is the capacity of the first peer to finish
- $F/C_s$: bottleneck time of the system - the minimum amount of time it takes for one peer to download the whole file from the server

Peer $p_i$ can upload any data it has received to any other peers, at rates not exceeding $C_{pi}$ in total. A peer is "finished" when it has received all data in the file.

We choose rates to minimize the total time peers are on,

$$\sum_{i=1}^{N} t_i. \tag{1}$$

If the controller does not know the power of each peer, this is the best way to minimize their total energy consumption.

## III. Optimal strategies for small systems

The strategy which minimizes (1) takes many forms, depending on the relative capacities. In contrast, the strategy for minimizing the last finish time [4], [13] takes on two distinct forms, depending on how large the server capacity $C_s$ is relative to the sum of the peer capacities $C_{pi}$. The optimal strategy for the "min-min" finish times [5], [11] has $N$ cases, depending on the "multiplicity", defined as the number of peers which can finish at time $F/C_s$, which is the minimum time for the server to transmit the entire file and is called the "bottleneck time". The multiplicity is $M$ if and only if [14]

$$C_s \leq \sum_{i=1}^{M} \frac{C_i}{M-1} + \sum_{i=M+1}^{N} \frac{C_i}{M}. \tag{2}$$

In particular, when $M = N$, the optimal solution [4], [13] is for peer $p_i$ to receive data at rate $C_{pi}/(N-1)$ from the server, which it forwards to all peers. The remaining capacity $C_s - \sum_{i=1}^{N} C_{pi}/(N-1)$ is split evenly among the peers. Each peer finishes downloading at $\frac{F}{C_s}$ which gives

$$\sum_{i=1}^{N} t_i = \frac{NF}{C_s}. \tag{3}$$

This section will describe the optimal strategies for most of the cases with three or fewer peers. The approach to showing optimality is similar to that of [11], [14]: First, a lower bound on the minimum possible time will be derived by considering "cut set bounds", or bounds on the maximum amount of data that can be proved to a given set of peers in a given time. Then a strategy will be described which achieves this lower bound, and is hence optimal.

### A. Implicit Lower Bound

In order to find explicit lower bounds on the objective (1), we first derive an implicit bound in terms of the individual finish times.

First, note that for $p_i \leq N-1$, the $i$th peer to finish can send at rate at most $C_{pi}$, and cannot send after it turns off at time $t_i$. Moreover, the last peer to finish can send at rate at most $C_{pN}$, and has no destination left to send to after the second last peer finishes receiving data at time $\tau_{N-1}$. Finally, the server can send at rate at most $C_s$, and cannot usefully send once all peers have finished downloading at time $t_N$. Yet all $N$ peers must receive the entire file of size $F$, whence

$$\sum_{i=1}^{N-1} t_i C_{pi} + \tau_{N-1} C_{pN} + t_N C_s \geq NF \tag{4}$$

or equivalently

$$t_N \geq \frac{NF - \tau_{N-1} C_{pN} - \sum_{i=1}^{N-1} t_i C_{pi}}{C_s}. \tag{5}$$

Adding $\sum_{i=1}^{N-1} t_i$ to both sides of (5), and noting that $\tau_N \leq t_N$ since peer $N$ must finish before it turns off, gives

$$\sum_{i=1}^{N} t_i \geq \frac{NF + t_{N-1}(C_s - C_{pN-1} - C_{pN})}{C_s} + \frac{\sum_{i=1}^{N-2} t_i(C_s - C_{pi})}{C_s}. \quad (6)$$

We will call (6) the general expression for the lower bound for the sum of on time of peers.

The following sections derive explicit lower bounds from (6) in terms of the size of the file and the capacities, and describe strategies which achieve those bounds.

*B. Networks of 1 or 2 Peers*

Trivially, a network with a single peer requires the peer to stay on until it receives the complete file at time $F/C_s$.

Similarly, if there are $N = 2$ peers with $C_s \leq C_1 + C_2$ then the multiplicity is $M = 2$, and by (3), $t_1 + t_2 \geq \frac{2F}{C_s}$.

The first non-trivial case is for $N = 2$ peers with $C_s > C_1 + C_2$. In this case, consider the following strategy, called Strategy A:

(i) Select an arbitrary order $p$ for the peers to finish downloading.

(ii) During $[0, t_1]$, the server sends disjoint data to peers $p_1$ and $p_2$ at rates $C_s - C_{p2}$ and $C_{p2}$ respectively. Peer $p_1$ uploads to peer $p_2$ at rate $C_{p1}$ and peer $p_2$ uploads to peer $p_1$ at rate $C_{p2}$. Peer $p_1$ turns off at $t_1 = \tau_1$, when it obtains the whole file.

(iii) During $[t_1, t_2]$, the server sends to peer $p_2$ at rate $C_s$.

**Proposition 1.** *When $N = 2$ and $C_s - C_1 - C_2 > 0$, Strategy A optimizes (1). No strategy with $t_{p1} > F/C_s$ can optimize (1).*

*Proof:* Substituting $t_1 \geq F/C_s$ into (6) gives

$$\sum_{i=1}^{2} t_i \geq \frac{F}{C_s}(3 - \frac{C_{p1} + C_{p2}}{C_s}) \quad (7)$$

with equality only if $t_{p1} = F/C_s$.

Under Strategy A, the peers turn off at

$$t_1 = \tau_1 = \frac{F}{(C_s - C_{p2}) + C_{p2}} = \frac{F}{C_s}. \quad (8)$$

$$t_2 = \tau_2 = t_1 + \frac{F - F_2(t_1)}{C_s}, \quad (9)$$

$$= \frac{F}{C_s} + \frac{F - \frac{F}{C_s}(C_{p1} + C_{p2})}{C_s}, \quad (10)$$

$$= \frac{F}{C_s}(2 - \frac{C_{p1} + C_{p2}}{C_s}). \quad (11)$$

Adding these gives (7) with equality, and so Strategy A minimizes the sum of on time of peers, and no strategy with $t_{p1} > F/C_s$ can. ■

Note that, unlike in the min-min problem [5], [11], the order of finishing the peers makes no difference to the sum of turn-off times in this case.

*C. Network of 3 Peers*

When there are three peers, there are at least four cases to consider.

The simplest case is again when all $M = N$ can finish by time $F/C_s$, namely $C_s \leq (C_1 + C_2 + C_3)/2$. In this case, the minimum cost is $3F/C_s$, by (3).

The case $M = 2$ can be split into two cases: The case when $(C_1 + C_2 + C_3)/2 < C_s \leq C_1 + C_2$ is still open, and the remaining one is considered in the next section. The difficult case $M = 1$ is presented last.

*1) Multiplicity $M = 2$:* Consider now the subset of cases with $M = 2$ such that $(C_1 + C_2 + C_3)/2 < C_1 + C_2 < C_s$.

It will be shown that the following strategy, called Strategy B, is optimal.

(i) Set the finish order $(p_1, p_2, p_3)$ to be an arbitrary order that satisfies

$$C_{p2} + C_{p3} < C_s \leq C_{p1} + C_{p2} + C_{p3}/2. \quad (12)$$

(ii) During $[0, t_1]$: The server sends disjoint data to peer $i \in \{p_1, p_2\}$ at rate $\lambda C_i$, where

$$\lambda = \frac{C_s - C_{p3}/2}{C_{p1} + C_{p2}} \quad (13)$$

Peer $p_1$ sends this data to peer $p_2$ and vice versa. Both send a subset of this at rate $C_i - \lambda C_i$ to peer $p_3$.

The server sends data, disjoint from that sent to peers $p_1$ and $p_2$, to peer $p_3$ at rate $C_{p3}/2$, which peer $p_3$ forwards to both peers $p_1$ and $p_2$.

Peers $p_1$ and $p_2$ turn off when they receive the whole file, at $t_1 = t_2 = \tau_1 = \tau_2$.

(iii) During $[t_1, t_3]$: The server sends to peer $p_3$ at rate $C_s$.

Note that the order $(1, 2, 3)$, which finishes the fastest peer first, always satisfies (12). However, other orders such as $(3, 2, 1)$ may also satisfy it.

**Proposition 2.** *When $N = 3$ and $C_1 + C_2 < C_s \leq C_1 + C_2 + \frac{C_3}{2}$, Strategy B is feasible and minimizes (1). No strategy with $t_{p2} > F/C_s$ can minimize (1).*

*Proof:* To obtain a lower bound on the finishing time, substitute $C_s - C_{p2} - C_{p3} > 0$, $t_2 \geq F/C_s$, $C_s - C_{p1} > 0$ and $t_1 \geq F/C_s$ into (6) to get

$$\sum_{i=1}^{3} t_i \geq \frac{F}{C_s}(5 - \frac{C_{p1} + C_{p2} + C_{p3}}{C_s}) \tag{14}$$

which holds with equaity only if $t_{p2} = F/C_s$. It remains to show that Strategy B is feasible and achieves this bound with equality.

First note that Strategy B is feasible. In particular, the fact that the multiplicity is $M < 3$ implies $C_s - C_{p3}/2 > (C_{p1} + C_{p2})/2$, whence $\lambda > 1/2$. Similarly, $\lambda \leq 1$ since the multiplicity is $M > 1$. This implies peers 1 and 2 have enough distinct data to send, at a non-negative rate, to peer 3 in the first stage.

Under Strategy B,

$$t_1 = t_2 = \tau_1 = \tau_2 = \frac{F}{\lambda C_{p1} + \lambda C_{p2} + C_{p3}/2} = \frac{F}{C_s}. \tag{15}$$

and

$$t_3 = \tau_3 = \frac{F}{C_s} + \frac{F - \frac{F}{C_s}(C_{p1} - \lambda C_{p1}}{C_s} + \frac{C_{p2} - \lambda C_{p2} + C_{p3}/2)}{C_s}$$
$$= \frac{F}{C_s}(3 - \frac{C_{p1} + C_{p2} + C_{p3}}{C_s}). \tag{16}$$

since $\lambda C_{p1} + \lambda C_{p2} = C_s - C_{p3}/2$.

Summing these gives (14) with equality:

$$\sum_{i=1}^{3} t_i = \frac{F}{C_s}(5 - \frac{C_1 + C_2 + C_3}{C_s}). \tag{17}$$

Since $C_{p1} + C_{p2} + C_{p3} = C_1 + C_2 + C_3$, (17) achieves equality in (14) and Strategy B is optimal for (1). ∎

In this case, there may again be multiple optimal finishing orders. However, the next case shows that is not always so.

*2) Multiplicity $M = 1$:* In this case, $C_s > C_1 + C_2 + \frac{C_3}{2}$.

In the optimal strategy, the server sends just enough to $p_3$ to enable that peer to send continuously to the other two until they finish. Let

$$f(x, y) = 4 - \frac{x}{C_s} + \frac{(2C_s - C_1 - x)(C_s - C_1 - y)}{C_s(C_s + \frac{y}{2})}. \tag{18}$$

and define Strategy C as follows:

(i) If $f(C_2, C_3) < f(C_3, C_2)$, set the finishing order $(p_1, p_2, p_3) = (2, 1, 3)$; otherwise set the finishing order $(p_1, p_2, p_3) = (3, 1, 2)$.

(ii) During $[0, t_1]$, the server sends different file segments to $p_1$, $p_2$ and $p_3$ at rates $C_s - C_{p2} - r_3$, $C_{p2}$, and $r_3$ respectively, where

$$r_3 := \frac{2C_s - C_{p1} - C_{p2}}{2C_s + C_{p3}} C_{p3}. \tag{19}$$

Then $p_1$ uploads to $p_2$ at rate $C_{p1}$, $p_2$ uploads to $p_1$ at rate $C_{p2}$, $p_3$ uploads to $p_1$ at rate $r_3$, and to $p_2$ at rate $C_{p3} - r_3$. When $p_1$ obtains the whole file at $\tau_1$, it immediately turns off which means $t_1 = \tau_1$.

(iii) During $[t_1, t_2]$, $p_3$ continues to upload to $p_2$ the data it received from the server during $[0, \tau_1]$. The server uploads at full rate to $p_2$, and $p_2$ uploads at its full rate to $p_3$. When $p_2$ obtains the whole file at $\tau_2$, it immediately turns off which means $t_2 = \tau_2$.

(iv) During $[t_2, t_3]$, $p_3$ continues to receive the remainder of the file at rate $C_s$ from the server until it obtains the whole file.

The following proposition is proved in Appendix A

**Proposition 3.** *When $N = 3$ and $C_s > C_1 + C_2 + \frac{C_3}{2}$, under Strategy C, the sum of finish times is*

$$\sum_{i=1}^{3} t_i = \frac{F}{C_s} \min\Big(f(C_2, C_3), f(C_3, C_2)\Big) \tag{20}$$

*which is the minimum achievable. Moreover, no strategy in which $p_1$ turns off after $p_2$ can achieve this.*

## IV. Heuristic for large systems

The foregoing results show that the optimal strategies quickly become complex as the number of peers grows. However the number of clients downloading a file may be very large. To study this case, it is useful to consider a heuristic strategy based on the insights from the explicit solutions. These insights include: It seems optimal for a peer to turn off as soon as it has finished downloading; The optimal order is not necessarily decreasing order.

This gives rise to Strategy D:

(i) Given a finishing order $p$, calculate maximum number of peers that can finish at $F/C_s$, denoted

$$M_p = \min\left\{M : C_s \leq \sum_{j=1}^{M} \frac{C_{pj}}{M-1} + \sum_{j=M+1}^{N} \frac{C_{pj}}{M}\right\}. \tag{21}$$

(ii) On $[0, F/C_s]$, the server sends to $p_i$ at rate $C_{pi}/M_U$ for $i > M$, with the remaining capacity divided among $p_1$ to $p_{M_p}$ in proportion to $C_{pi}$. Each peer sends a copy of everything it receives to $p_1$ to $p_{M_p}$, except itself. Peers $p_1$ to $p_{M_p}$ turn off at $F/C_s$.

(iii) On $[\tau_{i-1}, \tau_i]$, for all $j > i \geq M_p + 1$, $p_j$ sends to $p_i$ at rate $C_j$; if it only has $D < (\tau_i - \tau_{i-1})C_j$ data and would not be able to send at rate $C_j$ for the entire time, then the server sends it new data at rate $C_j - D/(\tau_i - \tau_{i-1})$. The server's remaining capacity is sent to $p_i$. If $i < N$ then $p_i$ sends to $p_{i+1}$ at rate $C_{pi}$. Peer $p_i$ turns off at $t_i = \tau_i$.

This was evaluated by simulation for varying numbers of peers $N$, varying distributions of upload capacities $C_i$, and varying regimes for scaling $C_s$ with the size of the network. In each case, a 500 MByte file was distributed to up to $10^4$ peers with randomly chosen capacities with mean 10 Mbit/s. The mean of 100 runs of each test is plotted.

Three finishing orders were considered: descending order of capacities, which appears to minimize finish times; random; ascending, which keeps the more useful peers active as long as possible. These were compared with two reference strategies: "Simultaneous" [4], [7], [13] in which all peers finish simultaneously at the earliest possible time; and "Sequential No P2P", in which the server sends the complete file to each peer in order, after which the peer turns off. Note that much less energy would be used if the server could wake each peer when its turn came to download.

### A. Constant Server Capacity with varying $N$

Figure 1 shows the objective (1) normalized by the number of peers, for $C_s = 10$ Gbit/s[1] and $C_i$ Pareto distributed, with shape parameter 0.5. As expected, the Sequential scheme increases linearly with the number of peers $N$, while the P2P schemes scale more gracefully. Even with P2P, the on time increases slightly, since $C_s$ remains fixed even though the peers' total capacity grows in proportion to $N$. Naively turning peers off while keeping the same finishing order as in [6], [11], in descending order of capacities, increases the energy consumption by a factor of over 2 relative to Simultaneous, since the extra delay incurred by the last peers to finish outweighs the savings. Conversely, finishing the slow peers first and turning them off reduces energy consumption by a factor of about 2 relative to Simultaneous.

The distribution of capacities $C_i$ makes a modest difference to the finish times, as shown in Figure 2 when

[1]Recall that the server is a software house, not a PC.

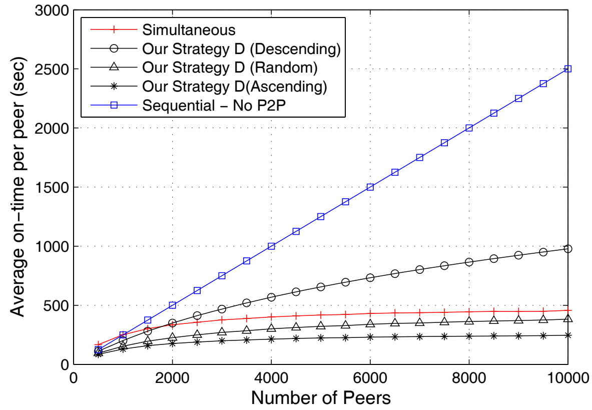


Figure 1. Average on-time per peer with constant $C_s = 10$ Gbps, $C_i$ is Pareto distributed with mean 10 Mbps

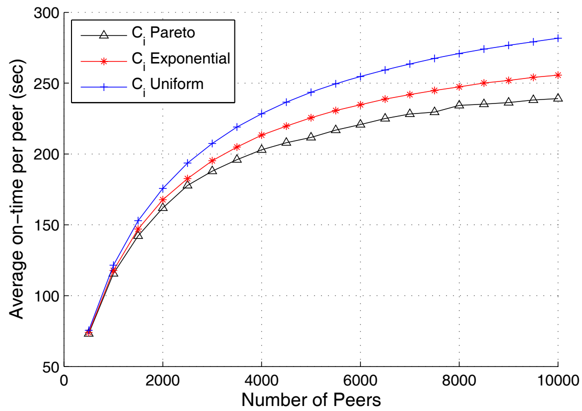


Figure 2. Average on-time per peer with constant $C_s = 10$ Gbps and different distributions of $C_i$, with mean 10 Mbps. $C_{pi}$ are ascending.

the $C_{pi}$ are an increasing permutation of $C_i$. When the $C_{pi}$ are a random permutation, there is no discernible dependence on the distribution. This is presumably because, in that case, the average capacity after $i$ peers have finished remains $10(N - i)$ Mbit/s regardless of the distribution. In contrast, for an increasing permutation, the average capacity decreases sublinearly in $i$, more slowly as the distribution becomes heavier tailed. This hypothesis is supported by the fact that the heavier tails give *longer* finish times when the $C_{pi}$ are a decreasing permutation.

### B. Varying Server Capacity

Popular content will typically be served by more powerful servers, and so it is reasonable to expect that the server capacity $C_s$ will increase with the number of peers $N$. However, it is unclear how it will scale.

One extreme is that $C_s$ may grow proportional to $N$.

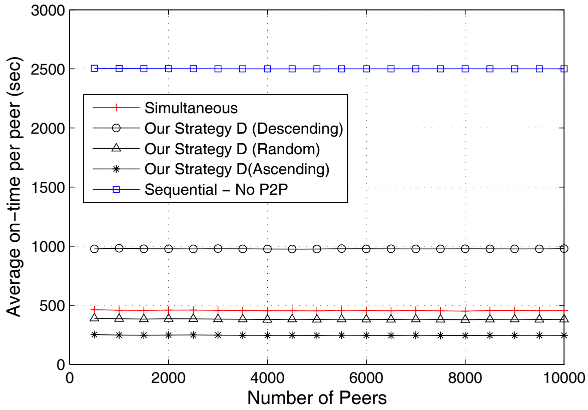


Figure 3. Average on-time per peer when $C_s = 0.1N$ Mbit/s,$C_i$ is Pareto distributed and $C_{pi}$ are ascending.

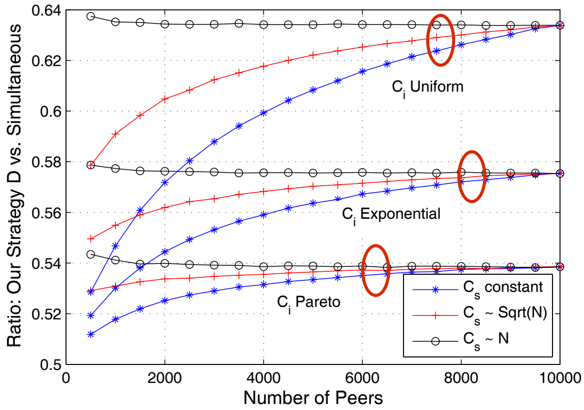


Figure 5. Ratio of Sum of peers' on-time: Our Strategy D vs. Simultaneous Strategy. $C_{pi}$ are increasing. The vertical axis does not start from 0.

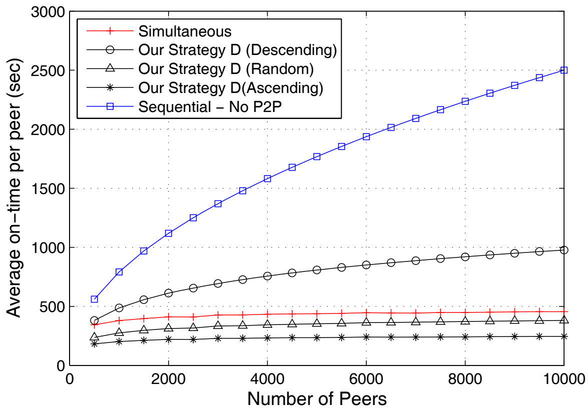


Figure 4. Average on-time per peer when $C_s = 10\sqrt{N}$ Mbit/s, $C_i$ is Pareto distributed and $C_{pi}$ are ascending.

Figure 3 shows that this results in the server staying on for a constant amount of time, regardless of how many peers there are, since both the amount of data and the capacity scale in proportion. The sleep-aware Strategy D still outperforms the others. The other extreme of constant $C_s$ was shown in Figure 1.

A less extreme scaling would be $C_s$ proportional to $\sqrt{N}$, as shown in Figure 4. In this figure, the capacity with $10^4$ peers equals that of Figure 1, and so the only change is an increase in on time for a small number of peers.

Figure 5 shows the average on-time per peer using our strategy D as a fraction of the time consumed by the Simultaneously strategy. In all cases of $C_s$, our strategy takes around half of the time taken using the Simultaneous strategy [2]. This matches the intuition that the $i$th peer is on for roughly $i/N$ of the time. The upward trend as $N$ increases for constant $C_s$ is because the peers provide more capacity as $N$ increases, and the penalty for turning them off becomes larger. The apparent convergence of all scalings of $C_s$ as $N$ increases is an artefact of the parameters, which make $C_s$ coincide for $N = 10^4$.

## V. Conclusion

Optimised peer to peer systems can substantially improve the energy efficiency of file distribution compared with systems in which end systems are powered on from the time the file becomes available until the time it is downloaded. For small networks, it can be shown rigorously that the optimal strategy turns each peer off as soon as it has finished downloading. It is tempting simply to take a system such as [11] which provides good download times, and to turn off peers when they finish downloading. However, this actually increases energy consumption. Instead, substantial savings are possible if the order of serving peers is altered so that higher-capacity peers remain in the system longer.

This work considered only an idealized model, but opens the way for many further studies. Apart from the natural tasks of considering download constraints, finding a decentralized solution and considering peers that arrive part way through transmission, which also apply to studies such as [5], [6], [10], [11], there are some extensions specific to energy efficiency. The model

[2]This comes with a $\sim 10\%$ increase in delay using our strategy D compared to the one which optimised delay.

could be expanded to include energy consumption of the server, including the non-linear relationship between its energy consumption and capacity. It could also consider peers with known, different power consumptions; in that case, it is likely to be optimal to finish peers in increasing order of upload capacity per watt. Perhaps most importantly, it will be useful to study systems in which peers are not left on purely for P2P downloading; the optimal strategy may be very different when peers are only participating when they are already on for other purposes.

## ACKNOWLEDGEMENT

This work was supported by Australian Research Council (ARC) grants DP0985322 and FT0991594.

## REFERENCES

[1] Australian Computer Society, *Carbon and Computers in Australia*, 2010.
[2] Y. Agarwal, S. Hodges, R. Chandra, J. Scott, P. Bahl, and R. Gupta, "Somniloquy: Augmenting network interfaces to reduce PC energy usage," in *Proc. Usenix NSDI*, 2009.
[3] J. Blackburn and K. Christensen, "A simulation study of a new green bittorrent," in *ICC Workshops*, 2009.
[4] J. Mundinger, R. Weber, and G. Weiss, "Optimal scheduling of peer-to-peer file dissemination," *J. Scheduling*, vol. 11, pp. 105–120, April 2008.
[5] M. Mehyar, G. WeiHsin, S. H. Low, M. Effros, and T. Ho, "Optimal strategies for efficient peer-to-peer file sharing," in *Proc. IEEE ICASSP*, 2007.
[6] Y. Wu, Y. C. Hu, J. Li, and P. A. Chou, "The delay region for P2P file transfer," in *Proc. Int. Symp. Info. Th. (ISIT)*, 2009, pp. 834–838.
[7] R. Kumar and K. Ross, "Peer-assisted file distribution: The minimum distribution time," in *Proc. IEEE HotWeb*, 2006.
[8] M. Lingjun and K.-S. Lui, "Scheduling in P2P file distribution – on reducing the average distribution time," in *IEEE Consumer Commun. Netw. Conf. (CCNC)*, Jan. 2008, pp. 521 –522.
[9] M. Lingjun, P.-S. Tsang, and K.-S. Lui, "Improving file distribution performance by grouping in peer-to-peer networks," *IEEE Trans. Netw. Serv. Manag.*, vol. 6, no. 3, pp. 149 –162, Sep. 2009.
[10] P.-S. Tsang, X. Meng, and K.-S. Lui, "A novel grouping strategy for reducing average distribution time in P2P file sharing," in *IEEE Int. Conf. Commun (ICC)*, May 2010, pp. 1 –5.
[11] G. Ezovski, A. Tang, and L. L. H. Andrew, "Minimizing average finish time in P2P networks," in *Proc. IEEE INFOCOM*, 2009.
[12] C. Chang, T. Ho, M. Effros, M. Medard, and B. Leong, "Issues in peer-to-peer networking: A coding optimization approach," in *IEEE Int. Symp. Network Coding (NetCod)*, Jun. 2010, pp. 1 –6.
[13] J. Mundinger, R. R. Weber, and G. Weiss, "Analysis of peer-to-peer file dissemination amongst users of different upload capacities," *SIGMETRICS Perform. Eval. Rev.*, vol. 34, pp. 5–6, September 2006.
[14] M. Mehyar, "Distributed averaging and efficient file sharing on peer-to-peer networks," Ph.D. dissertation, California Institute of Technology, 2006.
[15] S. Prajna, Papachristodoulou, and P. A. Parrilo, "Introducing SOSTOOLS: a general purpose sum of squares programming solver," in *Proc. CDC*, 2001.

## APPENDIX A
## PROOF OF PROPOSITION 3

First, note that Strategy C is feasible. In particular, since $C_{p3}/2 \le r_3 \le \min\{C_{p3}, C_s - C_{p1} - C_{p2}\}$, the server can allocate rate $r_3$ to $p_3$.

Next, note that it achieves (39). Peer $p_1$ finishes downloading and turns off at

$$t_1 = \frac{F}{(C_s - C_{p2} - r_3) + C_{p2} + r_3} = \frac{F}{C_s}. \tag{22}$$

Peer $p_2$ finishes downloading and turns off at

$$\begin{aligned} t_2 &= t_1 + \frac{F - F_2(t_1)}{C_s + C_{p3}} && (23)\\ &= \frac{F}{C_s} + \frac{F - \frac{F}{C_s}(C_{p1} + C_{p2} + (C_{p3} - r_3))}{C_s + C_{p3}} && (24)\\ &= \frac{F}{C_s}\left(\frac{2C_s - C_{p1} - C_{p2}}{C_s + C_{p3}/2}\right) && (25) \end{aligned}$$

Peer $p_3$ finishes downloading and turns off at

$$\begin{aligned} t_3 &= t_2 + \frac{F - F_3(t_2)}{C_s} && (26)\\ &= t_2 + \frac{F - F_3(t_1) - (t_2 - t_1)C_{p2}}{C_s} && (27)\\ &= \frac{F}{C_s}\left(\frac{2C_s - C_{p1} - C_{p2}}{C_s + C_{p3}/2}\right) + && (28)\\ &\quad \frac{F - \frac{F}{C_s}r_3 - \frac{F}{C_s}(\frac{C_s - C_{p1} - C_{p2} - C_{p3}/2}{C_s + C_{p3}/2})C_{p2}}{C_s}\\ &= \frac{F}{C_s}\left(3 + \frac{C_{p1}}{C_s} - \frac{(2C_s - C_{p2})(C_{p1} + C_{p2} + C_{p3})}{C_s(C_s + C_{p3}/2)}\right). && (29) \end{aligned}$$

The sum of on time of peers is then

$$\begin{aligned} \sum_{i=1}^{3} t_i = \frac{F}{C_s} + \frac{F}{C_s}\left(\frac{2C_s - C_{p1} - C_{p2}}{C_s + C_{p3}/2}\right) + && (30)\\ \frac{F}{C_s}\left(3 + \frac{C_{p1}}{C_s} - \frac{(2C_s - C_{p2})(C_{p1} + C_{p2} + C_{p3})}{C_s(C_s + C_{p3}/2)}\right), \end{aligned}$$

which can written in the form

$$\begin{aligned} \sum_{i=1}^{3} t_i = && (31)\\ \frac{F}{C_s}\left(4 - \frac{C_{p1}}{C_s} + \frac{(2C_s - C_{p1} - C_{p2})(C_s - C_{p2} - C_{p3})}{C_s(C_s + C_{p3}/2)}\right). \end{aligned}$$

By the choice of the order in which the peers are finished, this is equal to (20).

The remainder of this appendix will show that (39) is the optimum, using a series of lemmas established in the next appendix.

For $N = 3$, (6) becomes

$$\sum_{i=1}^{3} t_i \geq \frac{3F + t_2(C_s - C_{p2} - C_{p3}) + t_1(C_s - C_{p1})}{C_s}. \tag{32}$$

From the multiplicity theorem, it is impossible that $t_1 = t_2 = \frac{F}{C_s}$ and so (14) is loose. A tight bound will now be derived, using the following lemmas, proved in the appendix.

**Lemma 1.** *For $N = 3$ and $M = 1$,*

$$t_2 \geq \frac{2F - C_{p2}\tau_1 - C_{p1}\min\{t_1, t_2\}}{C_s + C_{p3}/2}. \tag{33}$$

Although we know that the download times satisfy $\tau_1 \leq \tau_2$, the order of the turn-off times $t_i$ depends on the chosen strategy. Two cases can be considered.

**Lemma 2.** *For $N = 3$ and $M = 1$, if $t_1 \geq t_2$ then*

$$t_2(C_s - C_{p2} - C_{p3}) + t_1(C_s - C_{p1}) \geq \frac{F}{C_s}\left(\frac{2C_s(2C_s - C_{p1} - C_{p2} - C_{p3})}{C_s + C_{p1} + C_{p2} + C_{p3}/2}\right). \tag{34}$$

If $t_1 \leq t_2$, then (33) becomes

$$t_2 \geq \frac{2F - C_{p2}\tau_1 - C_{p1}t_1}{C_s + C_{p3}/2}. \tag{35}$$

Multiplying both sides of (35) by $C_s - C_{p2} - C_{p3}$ (which is positive) and then adding $t_1(C_s - C_{p1})$ to both sides gives

$$t_2(C_s - C_{p2} - C_{p3}) + t_1(C_s - C_{p1}) \geq \frac{(C_s - C_{p2} - C_{p3})(2F - C_{p2}\tau_1)}{C_s + C_{p3}/2} + \frac{t_1[(C_s - C_{p1})(C_s + C_{p3}/2) - C_{p1}(C_s - C_{p2} - C_{p3})]}{C_s + C_{p3}/2}. \tag{36}$$

Finally, the RHS of (36) is minimized when $\tau_1$ is maximum. Since $\tau_1 \leq t_1$,

$$t_2(C_s - C_{p2} - C_{p3}) + t_1(C_s - C_{p1}) \geq \frac{2F(C_s - C_{p2} - C_{p3}) + Kt_1}{C_s + C_{p3}/2}. \tag{37}$$

where

$$K = (C_s - C_{p1})(C_s + \frac{C_{p3}}{2}) - (C_{p1} + C_{p2})(C_s - C_{p2} - C_{p3}) \tag{38}$$

The sign of $K$ depends on the strategy. We will show that choosing $K \leq 0$ gives a sub-optimal strategy.

**Lemma 3.** *For $N = 3$ and $M = 1$, then if $t_1 < t_2$ and $K > 0$ then, for $f$ given by (18),*

$$\sum_{i=1}^{3} t_i \geq \frac{F}{C_s}\min\Big(f(C_2, C_3), f(C_3, C_2)\Big) \tag{39}$$

**Lemma 4.** *For $N = 3$ and $M = 1$, if $t_1 < t_2$ and $K \leq 0$ then (34) holds again.*

The following lemma is proved using the sum-of-squares technique [15].

**Lemma 5.** *For $N = 3$ and $M = 1$, if (34) holds, then (39) holds with strict inequality.*

## Appendix B
## Proofs of Lemmas

*Proof of Lemma 1:* Since $p_1$ and $p_2$ have both finished at time $t = \tau_2$, at least $2F$ data must have been received. An upper bound of the sum of the amount of data entering $p_1$ and that entering $p_2$ by time $\tau_2$ is

$$C_s\tau_2 + C_{p1}\min\{t_1, \tau_2\} + C_{p2}\tau_1 + C_{p3}/2\tau_2 \geq 2F \tag{40}$$

The first three terms are the maximum contributions from the server, $p_1$ and $p_2$ respectively, since $p_1$ cannot send after it turns off at $t_1$, and $p_2$ cannot send to either $p_1$ or itself once $p_1$ has finished downloading. The final term, the contribution from $p_3$, is only $\tau_2 C_{p3}/2$ because $p_3$ can only forward data which it has already received, and the only way for $p_3$ to have received that data is for it to have been diverted from $p_1$ or $p_2$; if an amount of data $x$ is redirected to $p_3$ then it can send at most $\min(2x, C_{p3})$.

This implies

$$\tau_2 \geq \frac{2F - C_{p2}\tau_1 - C_{p1}\min\{t_1, \tau_2\}}{C_s + C_{p3}/2}. \tag{41}$$

Since $t_2 \geq \tau_2$, this implies (33). ■

*Proof of Lemma 2:* When $t_1 \geq t_2$, (33) becomes

$$t_2 \geq \frac{2F - C_{p2}\tau_1}{C_s + C_{p1} + C_{p3}/2}. \tag{42}$$

Multiplying both sides of (42) by $C_s - C_{p2} - C_{p3}$ (which is positive) and then adding $t_1(C_s - C_{p1})$ to both sides gives

$$t_2(C_s - C_{p2} - C_{p3}) + t_1(C_s - C_{p1}) \geq \frac{(2F - C_{p2}\tau_1)(C_s - C_{p2} - C_{p3})}{C_s + C_{p1} + C_{p3}/2} + t_1(C_s - C_{p1}). \tag{43}$$

Finally, the RHS of (43) is minimized when $\tau_1$ is maximum and $t_1$ is minimum. We know that $t_1 \geq t_2$

and that $\tau_1 \leq \tau_2 \leq t_2$. By using these bounds we can obtain a bound on (43) to be (34). ■

*Proof of Lemma 4:* Since $t_2 > t_1$,

$$\tilde{t} \equiv \frac{t_2(C_s - C_{p2} - C_{p3}) + t_1(C_s - C_{p1})}{2C_s - C_{p1} - C_{p2} - C_{p3}} > t_1.$$

Substituting this into (37), and using $K \leq 0$, gives

$$\tilde{t}(2C_s - C_{p1} - C_{p2} - C_{p3}) \geq \frac{2F(C_s - C_{p2} - C_{p3})}{C_s + C_{p3}/2} + \frac{\tilde{t}[(C_s - C_{p1})(C_s + C_{p3}/2) - (C_{p1} + C_{p2})(C_s - C_{p2} - C_{p3})]}{C_s + C_{p3}/2}$$

and equivalently

$$\tilde{t} \geq \frac{2F}{C_s + C_{p1} + C_{p2} + C_{p3}/2}.$$

Multiplying both sides by the positive quantity $2C_s - C_{p1} - C_{p2} - C_{p3}$, and substituting for $\tilde{t}$ gives (34). ■

*Proof of Lemma 3:* Substituting $t_1 \geq \frac{F}{C_s}$ into (37) gives

$$t_2(C_s - C_{p2} - C_{p3}) + t_1(C_s - C_{p1}) \geq \frac{F}{C_s}\left(C_s - C_{p1} + \frac{(2C_s - C_{p1} - C_{p2})(C_s - C_{p2} - C_{p3})}{C_s + C_{p3}/2}\right), \tag{44}$$

when $t_1 < t_2$ and $K > 0$.

Because of the minimization within (39), substituting (44) into (32) proves the lemma, provided the RHS of (44) is minimized when $C_{p2} = C_1$ (i.e., the largest capacity peer finishes second).

To prove that, we first show that $C_{p2} \geq C_{p3}$. Note that if the values of $C_{p2}$ and $C_{p3}$ are interchanged then the only part of (39) that varies is the factor

$$A = \frac{2C_s - C_{p1} - C_{p2}}{C_s + C_{p3}/2}. \tag{45}$$

For (45) to be minimized with respect to the selection of $C_{p2}$ and $C_{p3}$,

$$\frac{2C_s - C_{p1} - C_{p2}}{2C_s + C_{p3}} \leq \frac{2C_s - C_{p1} - C_{p3}}{2C_s + C_{p2}}$$

which is only true if $C_{p2} \geq C_{p3}$.

Finally, we use a similar approach to show that $C_{p2} \geq C_{p1}$. Therefore we want

$$A(C_s - C_{p2} - C_{p3}) - C_{p1} \leq A(C_s - C_{p1} - C_{p3}) - C_{p2},$$

which can be simplified to

$$0 \leq (A - 1)(C_{p2} - C_{p1}).$$

Since $A > 1$ for $M = 1$, this implies $C_{p2} \geq C_{p1}$. Since $C_{p2} \geq C_{p1}$ and $C_{p2} \geq C_{p3}$, then $C_{p2} = C_1$. ■

*Proof of Lemma 5:* Substituting (34) into (32) gives a lower bound on the sum of on time of peers to be

$$\sum_{i=1}^{3} t_i \geq \frac{F}{C_s}\left(\frac{7C_s + C_{p1} + C_{p2} - C_{p3}/2}{C_s + C_{p1} + C_{p2} + C_{p3}/2}\right). \tag{46}$$

Next, we can minimize the RHS of (34) by allocating capacity $C_{p3} = C_3$ (i.e. the smallest capacity peer to finish last).

From the RHS of (34), we can see that the numerator is constant regardless of the order of when peers finish however the same can't be said for the denominator. Therefore, the RHS of (34) is minimized when the denominator is maximized and this is clearly the case when $C_{p3}$, the third peer to finish is allocated $C_3$, the smallest capacity.

Therefore, a lower bound on the sum of on time of peers when the coefficient of $t_1$ in (37) is negative is

$$\sum_{i=1}^{3} t_i \geq \frac{F}{C_s}\left(\frac{7C_s + C_1 + C_2 - \frac{C_3}{2}}{C_s + C_1 + C_2 + \frac{C_3}{2}}\right). \tag{47}$$

It is sufficient that (47) is always greater than $(F/C_s)f(C_2, C_3)$, i.e.,

$$\left(\frac{7C_s + C_1 + C_2 - \frac{C_3}{2}}{C_s + C_1 + C_2 + \frac{C_3}{2}}\right) - \left(4 - \frac{C_2}{C_s} + \frac{(2C_s - C_1 - C_2)(C_s - C_1 - C_3)}{C_s(C_s + \frac{C_3}{2})}\right) > 0 \tag{48}$$

This can be proved using the sum-of-squares method [15]. Place the terms of (48) on a common (positive) denominator, and then let

$$D_3^2 = C_3$$
$$D_2^2 = C_2 - C_3$$
$$D_1^2 = C_1 - C_2$$
$$D_s^2 = C_s - C_1 - C_2 - C_3/2,$$

which reflect the positivity and ordering constraints of the problem. The numerator then becomes

$$(D_s^4)^2 + (D_s^2 D_1)^2 + (D_s^2 D_2)^2 + (\sqrt{\frac{5}{2}} D_s^2 D_3)^2 + (\sqrt{2} D_s D_1 D_2)^2 + (2 D_s D_1 D_3)^2 + (\sqrt{3} D_s D_2 D_3)^2 + (D_s D_1^2)^2 + (\sqrt{\frac{7}{2}} D_s D_3^2)^2 \tag{49}$$

which is strictly positive since $D_s > 0$. ■