

# On the Efficiency and Cost of Introducing QoS in BitTorrent

Nazareno Andrade

Jaindson Santana

Francisco Brasileiro

Walfredo Cirne

Universidade Federal de Campina Grande

{nazareno,jaindson,fubica,walfredo}@ourgrid.org

## Abstract

*BitTorrent is currently a de facto standard for scalable content-distribution. However, its peer-to-peer model for resource allocation does not provide high availability and its performance depends on best-effort contributions given by peers. This has motivated several content-providers to use a hybrid model in which they operate a superpeer in order to attain a higher quality of service. In this paper, we use BitTorrent traces and analytical modelling to investigate the cost incurred by such an entity in relation to the benefits it can provide to the system.*

## 1 Introduction

BitTorrent is arguably the most popular scalable content distribution mechanism presently. The efficiency of the protocol and some popular lightweight implementations of it account for its success as a significant step in democratizing content publishing on the Internet. BitTorrent offloads the content provider by leveraging the resources of content consumers during download.

Nevertheless, although BitTorrent allows content to be distributed cheaply, its typical use, which we call standard BitTorrent, relies completely on content consumers to provide both availability and download performance. As the contribution of peers varies, the quality of service may drop below acceptable levels. Guo et al. [7] have found that an average of 10% of all requests for a content object fail in a popular content-sharing community.

Although this quality of service might be sufficient for some settings, it is very low for the standards of traditional content distribution. The need to provide a more reliable service while taking advantage of the scalability and efficiency of BitTorrent has motivated a growing use of an alternative model which we dub a *hybrid* BitTorrent model. In such model, a content provider tries to increase availability and performance in its content distribution operating a reliable component. We dub such component a *superseeder*

and its resemblance to a server in a peer-to-peer context accounts for our nomenclature. The hybrid model is currently used to distribute software updates [13], in user-generated content-sharing communities [3, 2], is being experimented by the TV and film industry [4, 1] and is sold as a product by companies [6].

However, although appealing, the hybrid model is still poorly understood. It is unclear how effective it is for a content provider to introduce a superseeder with some capacity in the content distribution process. Furthermore, it is also not clear what is the cost for this content provider to use the hybrid model compared to providing an equivalent quality of service through a centralized alternative. In this study, we shed some light on these questions by modeling and analyzing the efficiency and cost of the hybrid model.

Our methodology is to create an analytical fluid model of the hybrid BitTorrent distribution and evaluate its behavior on different scenarios which we obtain from traces of real system usage. Our main contributions are the development of a model which one can use to estimate the cost and efficiency of operating a superseeder in a known setting and an assessment of this cost and efficiency under real conditions taken from a popular content-sharing community.

The rest of this paper is structured as follows. We first review related work in Section 2 and explain briefly how BitTorrent works in Section 3. We then present our model for the hybrid BitTorrent in Section 4. In Section 5 we use this model and data from a BitTorrent trace to analyze the cost and impact of operating a superseeder in BitTorrent. We discuss the implications of our analysis and make some final remarks on Section 6.

## 2 Related Work

A considerable amount of research has been conducted recently on BitTorrent through analysis [11, 7], simulation [12] and measurements [10, 8]. These studies collectively assess the scalability and effectiveness of the standard BitTorrent protocol.

Qiu and Srikant defined a fluid model to study BitTorrent [11] in which they assumed a Poisson arrival model for the arrival time of requests. Later, Powelse et al. [10] as well as Guo et al. [7] identified, based on traces of BitTorrent usage, that this assumption is not realistic. Guo et al. modeled the request arrival as an exponential function and adapted Qiu and Srikant’s fluid model to reflect that.

Guo et al. also found shortcomings on the BitTorrent distribution model: low content availability, high fluctuation of client performance and unfairness of contributions. The first two shortcomings are of interest to us. Guo et al. have proposed an extension of the BitTorrent protocol to tackle these limitations based on inter-torrent cooperation. In this work, however, we focus on another method to tackle this limitations which is currently in production: operating a superseeder in a hybrid BitTorrent model.

Also related to our work, Stutzbach, Zappala and Rejaie have modelled a swarming system and analyzed its effectiveness when compared to the plain client/server model [12]. Their study uses a conservative model and shows that swarming scales much better than a centralized server. We are interested in evaluating how this scalability is affected by introducing the superseeder.

### 3 BitTorrent

To download a file using BitTorrent, a user must join a torrent, which is the network formed by all peers taking part in the distribution of a content object at a given instant. Peers which have an incomplete copy of the object are called *leechers*, while peers which have finished downloading and are still in the torrent are called *seeders*.

To distribute a file using BitTorrent, a content producer creates a .torrent metadata file which describes the division of the original file in chunks and specifies a tracker which must be used to join the torrent of the file. This .torrent file is usually distributed through web servers and serves as an entrance point for the torrent.

When a client downloads this file and contacts the tracker, it receives a list of other peers it should connect to. It then starts to exchange chunks of the file with them. BitTorrent has a built-in incentive mechanism through which leechers reward each other for serving file chunks with higher rates. Seeders, however, only upload chunks, and there are no incentives for seeding.

### 4 Model

In this section we present the model for a hybrid BitTorrent system in which an entity operates a superseeder. Our model is an extension of that devised by Guo et al. [7] upon a previous one proposed by Qiu and Srikant [11].

We make one main simplifying assumption in this model: that all peers in a torrent have identical upload bandwidths and identical download bandwidths.

Legout et al. [9] studied experimentally the service peers with homogeneous and heterogeneous upload links get from a torrent and found that the download speed each peer gets is generally related to its upload speed. This suggests it is reasonable to assume that heterogeneous peers would get the average benefit predicted by our model with individual peers obtaining download rates weighted by their contribution to the torrent.

We now present the model for a standard BitTorrent and then we discuss the extension of this model to encompass the hybrid BitTorrent.

#### 4.1 Standard BitTorrent

Guo et al. [7] found that the peer arrival rate at a torrent is given by an exponential decreasing function of the time. Given an initial arrival rate of peers  $\lambda_0$  and a popularity attenuation parameter  $\tau$ , the arrival rate  $\lambda(t)$  at time  $t$  is given by  $\lambda(t) = \lambda_0 e^{-\frac{t}{\tau}}$ . From this function it is possible to find the total number of requests for the torrent  $N_{all} = \int_0^\infty \lambda_0 e^{-\frac{t}{\tau}} dt = \lambda_0 \tau$ .

We consider that after joining a torrent, a peer has some probability of giving up the download and leaving forever. If it does not give up, it stays online until it finishes downloading, seeds for some time and leaves the torrent forever. At any given time  $t$  there are  $x(t)$  leechers and  $y(t)$  seeders in a torrent. Leechers give up downloading with rate  $\theta$  and seeders leave the system with rate  $\gamma$ . Each peer has an upload bandwidth  $\mu$  and a download bandwidth  $c$ , with  $c \geq \mu$ . For the sake of simplicity, the model considers a file of size 1, what implies that both  $\mu$  and  $c$  must be normalized by the file size when considering a real system.

A seeder is always able to upload to a leecher. A leecher is able to upload to another leecher with some probability which is defined as the file sharing efficiency  $\eta$ . Qiu and Srikant have shown that for a file which is divided in  $P$  parts, if peers maintain connections to  $k$  other peers,  $\eta \approx 1 - (\frac{\log(P)}{P})^k$  [11].

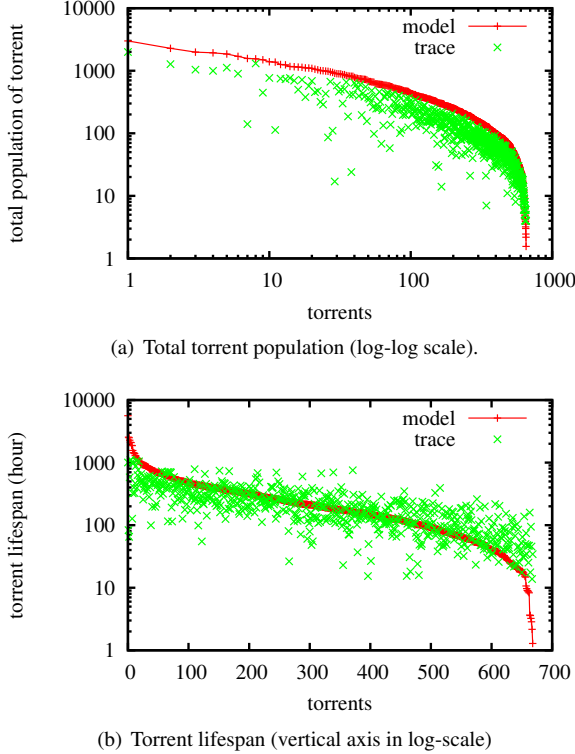
At each instant  $t$ , a number of peers finish downloading and become seeders. As the file size is 1, this number is given by the amount of bandwidth provided by peers at time  $t$ . We denote this amount  $\Phi(t)$ , which is the minimum between all upload bandwidth made available by all peers and the total download bandwidth of leechers, defined as  $\Phi(t) = \min\{\mu(\eta x(t) + y(t)), cx(t)\}$ .

We can express the rate of change in  $x(t)$  and  $y(t)$  using  $\lambda(t)$ ,  $\theta$ ,  $\gamma$  and  $\Phi(t)$  through the following set of ordinary differential equations:

$$\begin{cases} \frac{dx(t)}{dt} = \lambda_0 e^{-\frac{t}{\tau}} - \theta x(t) - \Phi(t), \\ \frac{dy(t)}{dt} = \Phi(t) - \gamma y(t), \\ x(0) = 0, y(0) = 1. \end{cases} \quad (1)$$

This set of equations is slightly different from that presented by Guo et al. because we assume  $c \geq \mu$ , instead of  $c \gg \mu$ .

To assess the capacity of the model of predicting reality, we get the parameters for the model through linear regression from the Umass BitTorrent trace [5] similarly to [7] and compare the torrent population and lifespan got in the traces with those predicted by the model. Figure 1 shows this comparison for the torrents which start and finish within the trace time frame. Further analysis of the accuracy of the model has been performed by Guo et al. [7].



**Figure 1. Comparison of model prediction and trace for the population and lifespan of torrents.**

## 4.2 Hybrid Model

We now consider that an entity is able to provide an amount of bandwidth  $\zeta(t)$  at time  $t$  to the leechers in a torrent. This can be done through one or more superseeders which act as seeders and are always available. For the sake

of simplicity, in the remainder of the paper we refer to a single superpeer, which is analogous to a group of superpeers with the same total capacity. The superseeder acts a seeders and is always online. The bandwidth provided by this peer is used as a complement to that already available.

As we consider a file of size 1, the amount of leechers which turn into seeders at time  $t$  is  $\Phi(t) + \zeta(t)$  and the set of ordinary differential equations which describes the hybrid model is:

$$\begin{cases} \frac{dx(t)}{dt} = \lambda_0 e^{-\frac{t}{\tau}} - \theta x(t) - \Phi(t) - \zeta(t), \\ \frac{dy(t)}{dt} = \Phi(t) - \gamma y(t) + \zeta(t), \\ x(0) = 0, y(0) = 0. \end{cases} \quad (2)$$

Note that  $y(t)$  is the amount of seeders besides the superseeder at time  $t$ . We assume the superseeder has the content being distributed in the start of the torrent and there is no other seeder at this time, so  $y(0) = 0$ .

We consider that the superseeder is able to provide a maximum total bandwidth  $b$  at a given instant. As  $\zeta(t)$  is provided by the superseeder only if there is spare download bandwidth among leechers, it is defined as:

$$\zeta(t) = \min\{b, cx(t) - \Phi(t)\} \quad (3)$$

The total data transfer performed by the superseeder is  $D = \int_0^\infty \zeta(t) dt$  and the average downloading speed of peers at time  $t$  in the hybrid model is  $u(t) = \frac{\Phi(t) + \zeta(t)}{x(t)}$ .

The download time  $\delta(t)$  for a peer which joins the torrent at time  $t$  and does not relinquish downloading is  $\delta = t' - t$  such that  $\int_t^{t'} u(t) dt = 1$ . We note  $\bar{\delta}$  for the average download time of all peers which can be found through the following integration:

$$\bar{\delta} = \frac{\int_0^\infty [\lambda(t) - \theta x(t)] \delta(t) dt}{\int_0^\infty [\lambda(t) - \theta x(t)] dt}. \quad (4)$$

## 5 Analysis

We now turn to evaluate the cost and efficiency of operating a superseeder to increase the quality of service in a torrent. As our set of equations has eight independent variables, it does not yield an elegant solution and we opted to analyze its results at points numerically instead of symbolically.

In the rest of this section, we first explain the metrics and the values for the parameters we use to then evaluate the model under different scenarios.

### 5.1 Metrics

We use the average download time  $\bar{\delta}$  in a torrent as our metric to evaluate the efficiency of the hybrid model.

As a measurement of cost, we compare cost of the distribution of a content object using a superseeder with that of using a centralized server to provide the same average quality of service for a population with the same dynamics as that in the torrent. In this case, we compare the maximum bandwidth  $B_c$  needed by a centralized server with the maximum bandwidth  $B$  used of the superseeder and the total data transfer  $D_c$  done by the centralized server with the total data transfer  $D$  performed by the superseeder. Note that the maximum bandwidth used of the superseeder might be lower than the total bandwidth it has available  $b$ .

We obtain  $B$  and  $D$  numerically from our model and  $B_c$  and  $D_c$  analytically as follows. Considering requests arrive with rate  $\lambda(t) = \lambda_0 e^{-t/\tau}$  and the centralized server is able to provide a download speed  $q \leq c$  for each requester, we model the variation in the number of downloaders  $w(t)$  at time  $t$  as:

$$\begin{cases} \frac{dw(t)}{dt} = \lambda_0 e^{-\frac{t}{\tau}} - \theta w(t) - qw(t), \\ w(0) = 0. \end{cases} \quad (5)$$

From which we find that  $w(t) = \frac{\lambda_0 \tau (e^{-\frac{t}{\tau}} - e^{-(\theta+q)t})}{\tau(\theta+q) - 1}$ .

The function  $w(t)$  assumes its maximum when  $t = t' = \tau \cdot \log((\theta + q)\tau) / ((\theta + q)\tau - 1)$ . Thus, the maximum instantaneous bandwidth needed by a server to provide a download speed  $q$  for each of its users under this circumstances is  $B_c = qw(t')$ .

The total data transfer from the centralized server, on the other hand is

$$D_c = q \int_0^\infty w(t) dt = \lambda_0 \tau \left( \frac{q}{\theta + q} \right). \quad (6)$$

Both  $B_c$  and  $D_c$  are normalized by the file size in these expressions.

Note that the performance of the standard BitTorrent with 100% availability is equivalent to the hybrid model with  $b = \mu$  and the cost of the standard model is always zero to the content provider.

## 5.2 Parameters

To analyze a scenario with the hybrid model, we need to instantiate eight variables. Naturally, a parameter sweep of all factors is not sensible. There are ranges of values for the variables which are unrealistic and there might be relationships among the variables which make certain combinations of them unlikely to be seen in practice.

We therefore use the trace of a real system to obtain representative combinations of the parameters which happen in standard BitTorrent to use in our analysis. We use the Umass trace [5], which consists of over 500 fully traced torrents over four months.

We get  $\lambda_0$  and  $\tau$  for each torrent through linear regression similarly to the process described in [7]. The parameters  $\gamma$  and  $\theta$  are given by the reciprocal of the average seeder service and time between quits of downloaders for each torrent, respectively. The file sizes are obtained directly from the description of torrents in the trace.

A limitation of the trace we used is that it only reports the download and upload rates of a peer accumulated over all torrents it is taking part in. It is not possible to derive the amount of bandwidth a peer devotes to each of these torrents. We opted to capture the ratio between download and upload rate and estimate values for one of them. For that, we computed the average ratio between download and upload bandwidth for peers in the trace<sup>1</sup> and found it to be 2.7. We then assumed an upload bandwidth of 10 KB/s for all peers, from which we got a download bandwidth of 27 KB/s.

We used hierarchical clustering to group our observations of  $\lambda_0$ ,  $\tau$ ,  $\gamma$ ,  $\theta$  and  $f$  accordingly to their similarity, from which we got 7 significant clusters. For each cluster, we select as a representative observation the torrent which has the smallest Euclidean distance to the point represented by the mean values of all observations in the cluster. We describe the clusters in terms of the 95% confidence intervals for the means of their parameters and of the representative values chosen for these parameters in Table 1. For presentation purposes, we also name each cluster accordingly to its most distinguishing characteristic.

The file sharing efficiency  $\eta$  is obtained from the file size as defined in Section 4. In BitTorrent, a file is typically divided in 256 KB pieces, and we consider peers maintain connection only to one other peer as a pessimistic boundary for our analysis.

Finally, we instantiate different values of  $b$  to investigate the effect of varying it in our metrics.

## 5.3 Evaluation

We look at the effect of varying the quality of service provided by the superseeder in our metrics for all torrents which we defined as representative in Table 1. We vary the bandwidth available to the superseeder so that  $b \cdot f$  varies between 10 KB/s – that available to the other peers – and 600 KB/s.

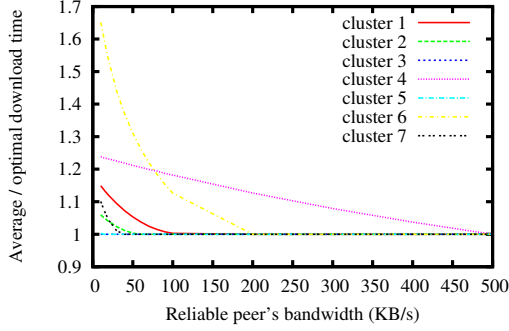
Figure 2 shows the ratio of  $\bar{\delta}$  over the optimal download time of peers ( $1/c$ ) when  $b \cdot f$  varies. Each line on the graph shows the behavior of a torrent, and is named accordingly to the cluster this torrent represents.

We can see that for most torrents  $\bar{\delta}$  decreases exponentially as  $b$  grows. This is because increasing  $b$  implies

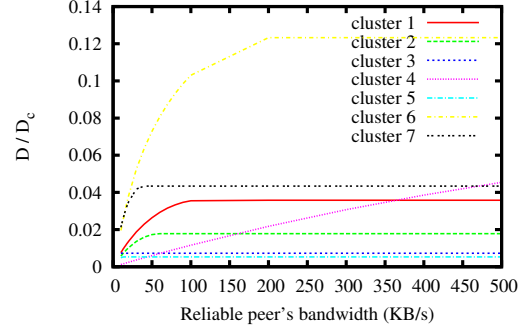
<sup>1</sup>Among which we filtered only peers which were both uploading and downloading at least 3 KB/s, so as to avoid including peers whose download or upload rate were strongly limited by the state of the torrent they were participating into.

**Table 1. Clusters of Torrents. For each variable, the representative value chosen and the 95% confidence interval for its mean.**

cluster	torrents	$\lambda_0 \cdot 10^2 (s^{-1})$	$\tau$ (days)	$\gamma \cdot 10^5 (s^{-1})$	$\theta \cdot 10^5 (s^{-1})$	$f$ (MB)
1-AverageCase	344	0.07 / [0.07;0.08]	4.71 / [3.72;5.23]	2.16 / [1.84;2.04]	1.6 / [1.51;1.72]	698.0 / [746.27;796.95]
2-AverageCase+Smaller-f	261	0.12 / [0.13;0.17]	2.12 / [2.94;3.97]	2.29 / [2.6;2.9]	3.01 / [3.61;4.32]	212.18 / [173.1;211.66]
3-Small-f+High- $\theta$ , $\gamma$	11	0.04 / [0.07;0.34]	0.83 / [0.33;0.86]	6.29 / [5.99;6.95]	19.04 / [17.11;22.23]	34.69 / [11.41;35.94]
4-Large- $\lambda_0$ +Small- $\tau$	6	1.27 / [1.29;1.54]	0.94 / [0.7;1.37]	1.74 / [1.7;3.27]	3.43 / [1.95;6.35]	636.13 / [104.13;650.96]
5-Small-f+High- $\theta$ +Small- $\tau$	34	0.11 / [0.11;0.16]	1.59 / [0.77;1.52]	3.51 / [3.02;3.74]	18.21 / [17.42;20.46]	52.91 / [33.6;85.48]
6-Largest-f	3	0.02 / [0.01;0.04]	10.62 / [2.28;10.62]	0.62 / [0.62;1.34]	0.51 / [0.51;0.7]	4096.0 / [3072.0;4096.0]
7-Small-f+High- $\gamma$ +Small- $\tau$	2	0.29 / [0.07;0.29]	0.22 / [0.09;0.22]	9.17 / [9.17;15.87]	2.16 / [0.72;2.16]	49.88 / [5.46;49.88]



**Figure 2. Performance evaluation of the hybrid model varying  $b$ .**



**Figure 3. Comparison of total data transfer in the hybrid and centralized models varying  $b$ .**

in seeders being created earlier, which changes the rate at which the coming leechers are turned into seeders.

On the other hand, when the initial rate of arrival of peers  $\lambda_0$  is very high,  $b$  becomes less effective in reducing the average download time. We can see that for the torrent of cluster 4,  $\bar{\delta}$  decreases approximately linearly as  $b$  increases.

For the torrents which represent cluster 3 and 5 and have small  $f$  values,  $\bar{\delta}$  is very close or equal to the optimal even for small values of  $b$ . Although the torrent of cluster 7 is similar to those of clusters 3 and 5, it performs 10% worse for small values of  $b$ . This is due to the lower service its seeders provide (represented by a much higher  $\gamma$  value).

For torrents with higher values of  $f$ ,  $\bar{\delta}$  needs higher values of  $b$  to reach the optimal value. This is explained by the fact that, for a given  $b$ , leechers take a time which is proportional to the file size to become seeders. The longer this takes to happen, the longer the system takes to reach its maximum throughput, and the more peers will be downloading with less than their full download speed.

Next, we look at how the cost of operating the super-seeder behaves when  $b$  varies. For each value of  $b$  in each torrent we calculate  $B$ ,  $B_c$ ,  $D$  and  $D_c$  and observe  $B/B_c$  and  $D/D_c$ . These represent the comparison of the cost of operating the hybrid and a fully centralized model which provide a same  $\bar{\delta}$  to an identical population of requesters.

Figure 3 plots  $D/D_c$  for different values of  $b$  and for

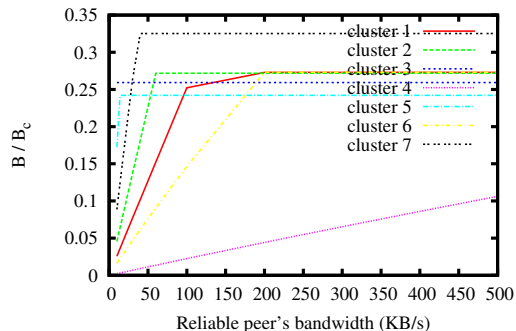
each of the 7 representative torrents. We can see that this value grows sub-linearly for most torrents and are limited by the point where there is a larger offer of upload bandwidth than demand for it. Again, an exception to these behaviors are the torrent of cluster 4 – with an approximately linear growth of  $D/D_c$  – and of clusters 3 and 5, where  $D/D_c$  is constant. The reasons are similar to those which explain the behavior of  $\bar{\delta}$ : the initial arrival rate of peers in cluster 4 is much higher than in the other torrents and there is always more bandwidth available than demanded for it in clusters 3 and 5.

The maximum value of  $D/D_c$  for the torrents we have examined is ordered similarly to the file size of the torrents. The exception is the torrent of cluster 7, which shows a high value of  $D/D_c$  compared to the other torrents. This denotes a smaller saving in data transfer and is due to its low seeder service, evidenced by a large value of  $\gamma$ .

Furthermore, it is worth mentioning that all values for  $D/D_c$  in the scenarios we analyzed were smaller than 0.14, and most were smaller than 0.05, assessing a great saving in the volume which needs to be transferred by a content provider which opts for the hybrid model in place of a centralized one.

Finally, we look at  $B/B_c$  for the scenarios we evaluate. Figure 4 shows this value for our different torrents when  $b$  varies. Recall that  $B$  is limited to the maximum amount

of bandwidth peers can simultaneously consume, which is smaller than that available for a high enough value of  $b$ . Also, note that as  $b$  grows linearly and  $\bar{\delta}$  decreases exponentially in the scenarios we have observed,  $B_c$  increases logarithmically. Therefore,  $B/B_c$  increases approximately linearly up to a point where  $\bar{\delta} = 1$  and  $B$  stops growing.



**Figure 4. Comparison of maximum bandwidth used in the hybrid and centralized models varying  $b$ .**

Interestingly, the largest saving in the maximum bandwidth needed to provide a certain quality of service happens for the torrent which represents cluster 4. Its most distinguishing characteristic is having a very large number of requests. In this case, although the performance of peers is not highly affected by  $b$ , the use of peers' resources makes the demand for bandwidth much lower for the superseeder than for the centralized alternative even for an optimal  $\bar{\delta}$ .

Again, the savings in the torrent which represents cluster 7 are smaller than those in the other torrents when comparing  $B$  and  $B_c$ . This happens because the high value of  $\gamma$  in this torrent implies in less resources from the peers and a higher load on the superseeder.

Another interesting observation is that most sets of parameters which happen in practice yield similar savings in the maximum bandwidth they demand to provide an optimal  $\bar{\delta}$ . For all representative torrents except those of clusters 4 and 7,  $B/B_c$  is between 0.22 and 0.75 when  $\bar{\delta} = 1$ .

It is also possible to compare  $D/D_c$  and  $B/B_c$  and see that for providing a certain quality of service, the savings in the hybrid model are smaller for the maximum bandwidth than for the total amount of data which the content provider will provide.

## 6 Conclusion

BitTorrent has proven to be a highly-scalable and efficient content distribution mechanism. However, when relying completely on the resources provided by content consumers, it fails to provide quality-of-service guarantees.

In this study we have investigated a method to circumvent this limitations which is gaining growing popularity: operating a superseeder to assist content distribution. We have presented a model which can be used by a content provider to analyze its costs and efficiency when operating such a model.

We have also analyzed the cost and efficiency of operating a superseeder with varying capacity in the scenarios derived from real BitTorrent usage. From this analysis it is possible to conclude that the hybrid model is still considerably cheaper than its centralized equivalent while able to provide near optimal performance with limited resources. The efficiency and savings were affected significantly by the seeding behavior of a class of torrents which we identified, however very low seeding was uncommon in the trace as a whole.

## Acknowledgements

This work was partially developed in collaboration with HP Brazil R&D.

## References

- [1] BitTorrent Company website: <http://www.bittorrent.com/2006-05-09-warner-bros.myt>. Warner Bros. Home Entertainment Group Announces Revolutionary Deal to Publish Legal Film and TV Content using the BitTorrent Platform, May 2006.
- [2] OurMedia: <http://www.ourmedia.org>, Dec. 2006.
- [3] Overmundo: <http://www.overmundo.com.br>, Dec. 2006.
- [4] BBC. Bbc imp homepage, 2006.
- [5] A. Bellissimo, P. Shenoy, and B. N. Levine. Exploring the use of bittorrent as the basis for a large trace repository. Technical Report 04-41, University of Massachusetts, June 2004.
- [6] M. Digital. Company website, 2006.
- [7] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang. Measurements, analysis, and modeling of bittorrent-like systems. In *Internet Measurement Conference 2005 (ACM SIGCOMM/USENIX IMC 2005)*, pages 19–21, Oct. 2005.
- [8] M. Izal, G. Urvoy-Keller, E. W. Biersack, P. Felber, A. A. Hamra, and L. Garcés-Erice. Dissecting BitTorrent: Five Months in a Torrent's Lifetime. In *PAM2004*, April 2004.
- [9] A. Legout, N. Liogkas, E. Kohler, and L. Zhang. Clustering and sharing incentives in bittorrent systems. Technical Report inria-00112066, version 1 - 21, INRIA, Nov. 2006.
- [10] J. A. Powelse, P. Garbacki, D. H. J. Epema, and H. J. Sips. Measurement study of the bittorrent peer-to-peer file-sharing system. Technical Report PDS-2004-003, Delft U. Technology, Apr. 2004.
- [11] D. Qiu and R. Srikant. Modeling and performance analysis of bittorrent-like peer-to-peer networks. In *SIGCOMM*, pages 367–378, Aug. 2004.
- [12] D. Stutzbach, D. Zappala, and R. Rejaie. The scalability of swarming peer-to-peer content delivery. In *NETWORKING*, pages 15–26, 2005.
- [13] Wikipedia. Blizzard downloader, 2006.