

A Performance Study of Peer-assisted File Distribution with Heterogeneous Swarms

Cristina Carbutaru, Yong Meng Teo, Ben Leong
Department of Computer Science
National University of Singapore
13 Computing Drive, Singapore 117417
[ccristina,teoy,bleong]@comp.nus.edu.sg

Abstract—Peer-to-peer file-sharing protocols, such as BitTorrent, have been widely used to improve the performance and scalability of file distribution systems. In this paper, we study the performance of peer-assisted file distribution systems with heterogeneous peers. Based on a measurement study of BitTorrent on PlanetLab, we made two key observations: (i) there is a fixed pattern in the utilization of the available bandwidth over the course of a download, and (ii) peers enjoy an amount of service that is commensurate with their contribution. Building on these insights, we developed an analytical model to estimate the download time for each class of peers in a well-provisioned peer-assisted file distribution system based on BitTorrent. Our model accurately predicts the download time and achieves an average error rate of 16.5% for heterogeneous swarms up to 150 nodes in size. We demonstrate how it can be used to estimate the server capacity for achieving a specific quality of service in a large heterogeneous swarm.

I. INTRODUCTION

As the size of the content distributed online increases, peer-to-peer (p2p) file-sharing protocols like BitTorrent (BT) have been adopted to augment traditional client-server file distribution systems. In such systems, the goal is to distribute files from a server to multiple clients in the shortest possible time. Peer-assisted systems improve the scalability and performance of content distribution because they utilize the upload bandwidth of the downloading clients to improve the overall available bandwidth of the system. However, distributed and uncoordinated p2p algorithms like BitTorrent are not entirely efficient in utilizing the available bandwidth of the system and allow clients to obtain service without offering equivalent service in return [19]. While managed architectures [18] and pricing mechanisms [17] have been proposed to address these issues, we believe that the popularity of BT and the availability of its numerous implementations make it an attractive choice for file distribution. Hence, there is a need to develop a better model for the performance of BT in the context of a peer-assisted file distribution architecture.

Previous work on p2p algorithms has focused mainly on systems in steady-state [11], [23]. While the steady-state assumption is reasonable for file-sharing systems since peers stay in the system and continue to share the file after they complete the download, this steady-state assumption may be not realistic for file distribution systems where peers download the file as

fast as possible and then leave. A *flash crowd*, where there is a sudden large surge in the number of users, often occurs when a new file is available for download. For a content distributor, the challenge is to ensure that the system has sufficient resources to cope with this sudden surge of users. Normal system behavior resumes when the system reaches steady-state.

To simulate the flash crowd scenario, we model the swarm as a closed system. Our goal is to predict the average download times for different classes of peers in a heterogeneous swarm. It is inherently difficult to model a heterogeneous system and such models are often difficult to solve in closed form. Nevertheless, we adopt a heterogeneous model because a homogeneous model is insufficient to accurately predict the characteristics of real systems [16].

In this paper, we analyze the performance of a heterogeneous bandwidth peer-assisted file distribution system based on BT. Our first observation from the measurement of PlanetLab [5] experiments is that bandwidth utilization is not constant over time and has a *step-like* profile towards the end of the download. Our second observation is that BT is relatively fair in distributing the upload capacity of the system among classes of peers with different upload bandwidths. From these two observations, we develop an analytical model for estimating the download time of each class of peers in a well-provisioned system.

Our analysis is based on the utilization of available bandwidth. In our previous work we showed that the utilization of available peer bandwidth, ρ , is less than one and is not constant over time for homogeneous p2p swarms in flash crowds [2]. We build on our earlier work by observing the bandwidth utilization of heterogeneous swarms running BT on PlanetLab [5]. We found that the evolution of ρ over time is characterized by three phases, which we term *startup*, *maximum utilization* and *end-game*. While the first two phases are similar to that for a homogeneous system, the last end-game phase has a step-like profile, with each step corresponding to the departure of the class of peers with the highest bandwidth. This behavior can be explained by the clustering phenomenon previously observed in BT systems [12]. We also show that clustering contributes to a proportional fair allocation of the available bandwidth to different classes of peers.

To complete our performance study, we develop an analytical model to estimate the average download time experienced by

each peer class by considering the utilization of available peer bandwidth. Furthermore, we capture in our model the effect of clustering in a heterogeneous swarm and analyze the impact of this phenomenon on the service received by each class of peers. We validate our model with measurements on PlanetLab and show that it is able to accurately estimate the average download times for different classes of peers, with an average error of 16.5%.

A main concern for file distribution systems is to provide sufficient server upload capacity such that clients achieve a minimum quality of service. Our model can be used to find the server capacity that strikes a balance between bandwidth costs and the specified download time experienced by peers.

The rest of the paper is organized as follows. In Section II, we present an overview of the related work. In Section III, we describe the key observations from the measurement results. Based on these observations, we derived an analytical model to estimate the download time of each class and validate it in Section IV. In Section V, we demonstrate how our model can be applied to estimate the required server capacity for achieving a desired quality of service. Section VI concludes this paper.

II. RELATED WORK

Unlike previous work that propose new centrally coordinated mechanisms [18] and new pricing mechanisms to incentivize uncoordinated p2p schemes [17], we study the effectiveness of using the popular BT algorithm directly for use in file distribution. While the performance of BT has been studied extensively as a file-sharing protocol [4], [10], [21], to the best of our knowledge, we are the first to directly study the performance of BT as a file distribution protocol. By file distribution, we mean that we provide a *server (BT seed)* as a constant source of data content for a set of clients that wish to download a file by collaborating according to the BT protocol.

Most previous work on the performance analysis of BT have also focused on homogeneous swarms [2], [11], [20], [23], [25]. The work on swarms with heterogeneous upload bandwidth typically models the systems at steady-state. While estimates for the average download time of the system have been proposed [10], [14], [16], the models are often difficult to solve for complex cases with more than three classes of peers [4], [6], [14] and it is hard to obtain the required parameters to do the estimates for real systems. On the other hand, the assumption of steady-state, with constant arrival rate, is often unrealistic in the context of file distribution because flash crowds are common [21], [25]. The few models that have been proposed to study flash crowds show that the protocol is scalable using branching processes [25] and that a steady-state will eventually be reached [20]. In contrast, we focus on modeling and validating with real measurements the performance of systems with flash crowds and estimate the download times for different number of classes.

In their work, Yang and Veciana [25] and Qiu and Srikant [20] used the measure of effectiveness of file-sharing, η , as an input parameter in their mathematical models. In

particular, they assume that a downloading peer's contribution to the service capacity is a fraction η of that of a peer that has fully downloaded the file and they assume that the total upload capacity of the peers is fully utilized at steady-state, i.e. $\eta = 1$. However, we observed that this assumption does not hold for a transient system, i.e. during a flash crowd. Our measurements on PlanetLab reveal that the utilization of available peer bandwidth is not a constant, but varies over time. We model the utilization of available peer bandwidth and use it to find the average download time expected by peers in different classes.

It has been shown that in well-provisioned BT swarms, peers tend to cluster with other peers that have similar upload bandwidth [12], [13], [16]. Moreover, the clusters of peers with higher upload bandwidths tend to contribute more to the swarm than lower capacity clusters [12]. Misra et al. highlighted that in a peer-assisted system, contributors should receive a fair price for the provided resources [17]. Similarly, we analyzed the service received by different classes of peers in a heterogeneous swarm and found that BitTorrent offers a reasonably fair share of the upload bandwidth to different classes of peers in the context of file distribution.

The impact of server capacity on the performance of homogeneous peer-assisted systems has been studied using fluid models [9], [22]. Various methods for server bandwidth allocation among different swarms and peers have been shown to improve performance both in the context of p2p streaming [24] and content distribution [3], [8]. Other proposed methods, such as content bundling [15] and dynamic allocation of peers among swarms [8], have been proposed to improve the download time and availability in p2p systems. Using our model, we analyze the impact of multiple classes of peers on the required server capacity to achieve a specific download time.

III. MEASUREMENT ANALYSIS

We investigated the performance of BitTorrent [7] as a file distribution protocol by conducting measurement experiments on PlanetLab [5]. In the process, we made two key observations: (i) there is a fixed pattern in the utilization of the available bandwidth over the course of a download, and (ii) peers enjoy an amount of service that is commensurate with their contribution.

In BT, the peers in a swarm cooperate to download large files, initially stored at a central location (seed), by simultaneously downloading and uploading different parts of the file from other peers, as well as directly from the seed. A file is divided into chunks, called *blocks*, and multiple blocks form a piece. A new peer connects to a tracker to obtain a list of active peers and their list of blocks. A peer downloads the first blocks from other peers and from the seeds. After the download is completed, BT peers can decide to stay in the swarm and become seeds, or leave the system. A mechanism called *choke/unchoke* regulates the exchange of blocks among peers, where each node attempts to upload blocks to the peers that offered it the best download rates during the last download interval. A number of

unchokes are chosen based on the best download rates, while one unchoke, called an *optimistic unchoke*, is randomly chosen from the pool of requests the peer received.

In our measurement study on PlanetLab, each experiment involves a tracker, a client that acts as the initial seed (which remains throughout the experiment) and clients that act as peers. The peers join the system at approximately the same time and a peer will leave the system immediately once its download is complete. This mimics a file distribution scenario where the clients are only interested in downloading a file and not in helping other clients with their downloads. Since the upload capacity of nodes on PlanetLab is unknown, we cap the upload bandwidth for different classes of peers and the seed using the default capping mechanism provided in BT to facilitate our analysis of the results. Because PlanetLab nodes are limited to uploading about 8 GB of data daily, we set the file size to 100 MB, and worked with swarms with up to 150 nodes and a maximum upload bandwidth of 256 kbps.

A. Utilization of Available Peer Bandwidth

Previous modeling work [20] claimed that the effectiveness of BT at utilizing available bandwidth can be approximated as one in steady-state. In our previous work [2], we found that for a homogeneous system in a flash crowd (transient state), the utilization of available bandwidth is not uniformly equal to one but varies with time.

Definition 1. Utilization of available peer bandwidth, ρ , is defined as the ratio of the effective upload bandwidth to the total upload capacity of peers in the system.

In Fig. 1, we plot the evolution of ρ over time for a heterogeneous system with 100 nodes divided equally between two classes on PlanetLab. The upload bandwidths of slow peers, fast peers and server are 64 kbps, 128 kbps and 256 kbps, respectively. We found that the bandwidth utilization in a heterogeneous system, similar to a homogeneous system [2], consists of three main phases, namely, startup, maximum utilization and

end-game. For this example, the startup phase is from 0 s to 100 s; the maximum utilization phase is from 100 s to 775 s, and the last (end-game) phase is from 775 s to 2500 s. After doing more than 100 experiments with different configurations on PlanetLab, we observed that during the maximum utilization phase, ρ is close to one, so BitTorrent is indeed a good protocol for file distribution.

The key difference between homogeneous and heterogeneous systems lies in the end-game phase. The nodes in a homogeneous swarm tend to finish their downloads and leave the system at approximately the same time. In a heterogeneous swarm, the end-game portion contains *steps* which correspond to the departure of the faster peers. In Fig. 1, we can see that a step occurs at around 1,000 s.

To better understand the utilization of the available system bandwidth as a download progresses, we represent the data in a slightly different form by plotting ρ as a function of K , the total number of blocks downloaded in the system. This is shown in Fig. 2. Since the number of file blocks downloaded by the peers depends on the time elapsed from the start of the download and on the number of peers, we believe that K captures the evolution of the system better than time. If N is the total number of peers in the system and M is the number of blocks in the downloaded file, all the peers would have downloaded the file when K reaches MN . Therefore the total number of blocks, K , can be normalized by dividing it by NM .

In Fig. 3, we plot ρ against $K(t)$ for another experiment with 150 nodes and three different classes of peers with different upload bandwidths. In this experiment, the server capacity is 256 kbps, 30% of the peers are slow (64 kbps), 60% are medium (128 kbps) and 10% are fast (192 kbps). Note that the vertical lines in Figs. 2 and 3 correspond to the moments when the first peers from the fastest remaining class in the system leave the system. In Fig. 3, we observed two sub-phases, each corresponding to peers from one class leaving the system. When 59% out of the total number of blocks are downloaded in the system, the fast peers start leaving the system. Later, when 75% of the blocks have been downloaded, the nodes with medium

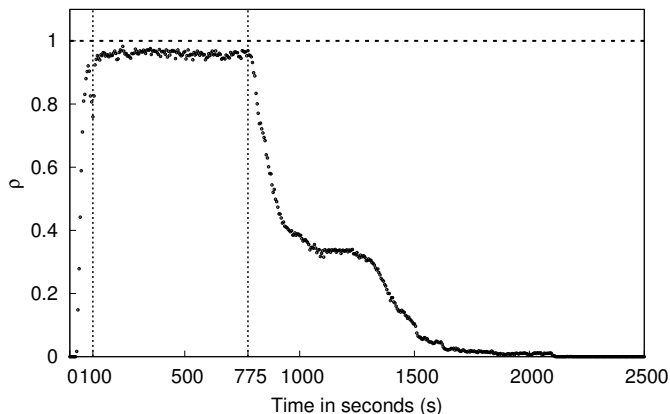


Fig. 1. Plot of ρ against time, t , for 100-node BT swarm with two classes.

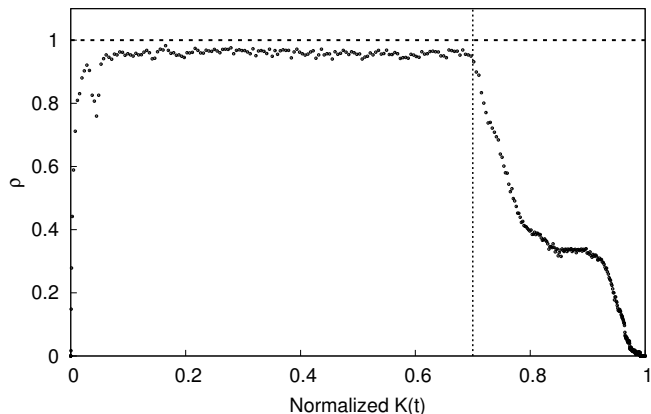


Fig. 2. Plot of ρ against $K(t)$ for 100-node BT swarm with two classes.

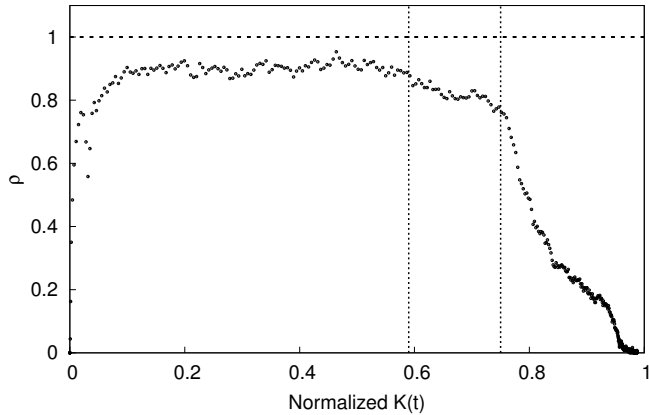


Fig. 3. ρ against $K(t)$ for 150-node BT swarm with three classes.

upload capacity start leaving the system, leaving only the slow peers.

These observations suggest that the end-game phase can be approximated with a sequence of steps. We can explain these steps with the observation that peers tend to cluster with peers of similar upload bandwidths as highlighted by Legout et al. [12]. In the ideal case, when clustering is perfect, they will finish their downloads together and the steps would be clearly defined. The number of steps matches the number of different classes of peers and when they occur depends on the relative bandwidths of the peers. The steps are delimited by the points in time when peers from the fastest class finish their downloads and start to leave the system. The value of ρ for each step depends on the upload bandwidth of the peers remaining in the system. While we noticed in our experiments that the steps in the end-game phase are not clearly delimited, likely because of asymmetries caused by the choke/unchoke policy and differing network conditions among peers in the swarm, we show that we are able to use steps to approximate the system performance to good effect in Section IV-A.

B. Clustering and Sharing in File Distribution

Next, we investigated the impact of imperfect clustering on the fairness of service distribution to peer classes. Due to clustering, BT peers tend to upload blocks to other peers within the same class and offer less service to peers outside their class. Our analysis of the end-game phase for the systems reflected in Figs. 1 to 3 confirms that peers tend to cluster with other peers of similar bandwidth. The clustering is however imperfect because there is always a chance that a peer might optimistically unchoke a peer from another class.

Previous work shows that fast peers are the main contributors to system service, by uploading more data by volume than the slower peers throughout the download process [12], [16]. However, in the context of file distribution, the key question is whether they also enjoy an amount of service that is commensurate with their contribution. In our measurement study,

we observed that each class receives a percentage of service that is close to the percentage of service offered by that class. This insight allows us to conclude that BitTorrent achieves good fairness when used for file distribution.

In analyzing the contribution versus the service received by peers, it is important to distinguish between total system service from peer-contributed service. Total system service includes the server contribution, and slow peers that stay for a longer time in the system would tend to receive more data from the server over the total download period than faster peers. Therefore, we exclude the server contribution from the system service when we analyze the fairness of the distribution of the upload bandwidth among the various classes of peers.

In Fig. 4, we plot the service variation (in terms of upload and download rate) with the total number of blocks downloaded in the system, for an experiment with 140 peers equally divided into two classes with 64 and 128 kBps upload capacity. Since we have a closed system, the total service offered by peers is equal to the total service received, after excluding the server's contribution. Therefore, we normalize the cumulative upload rate of each class with the total upload rate of the remaining peers in the system. Similarly, the cumulative download rate of each class is normalized with the total download rate of the peers in the system. We plot these normalized upload and download values, called *normalized service*, against the normalized total number of blocks downloaded in the system, $K(t)$, in Fig. 4.

We observed that the service share received by each class is comparable with the service share offered by that class. The total upload service in the system is divided by the continuous line between the slow class (shaded area) and the fast class (white area) in Fig. 4. Similarly, the dotted line separates the download service received by the slow and fast classes. The “ideal” line for the normalized service is at 0.33 and increases sharply to 1 when the fast peers leave the system. The slow peers receive slightly more service than what they contribute to the system, while fast peers receive slightly less.

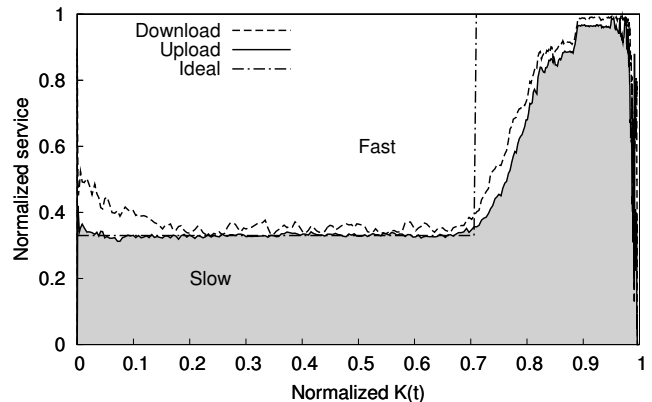


Fig. 4. Measured service enjoyed by the slow peers for swarm with 140 nodes and two classes.

This observation is consistent in all our experiments.

The ideal line represents perfect fairness, i.e. where the slow and fast peers contribute and receive service that is exactly equal to their upload capacity. The ideal service line is computed using the cumulative upload capacities of peers in one class over the total capacity of the system (excluding the server). We observe that the measured upload (contribution) matches the ideal line, but in terms of measured download (service), the slower peers consume slightly more than their fair share.

The fairness in service distribution is strictly related to the share ratio of each class [1], [10]. By share ratio, we refer to the fraction of offered service (upload) to the received service (download). If all peers received service that is equal to their contributions to the system, the share ratio would be one. In this work, we are interested not only in the share ratio for individual peers, but the share ratio for a class of peers.

Definition 2. The *class share ratio* for a class of peers is defined as the *ratio of the cumulative data uploaded by all the peers in that class to the cumulative data downloaded by these peers, excluding contributions from the server.*

In Fig. 5, we plot the class share ratio for the slow peers in a swarm with 100 peers with two classes of peers and a server capacity of 256 kbps as the proportion of slow peers (64 kbps) against fast peers (128 kbps) varies. Ideally, the share ratio should be one to ensure fairness between classes. Our results show that the slow peers achieve a share ratio smaller than one, though the share ratio increases when the proportion of slow peers increases. When the fraction of slow peers is small, the slow peers do get somewhat more service than their corresponding contributions to the system. Intuitively this leads to shorter download times for them when the proportion of slow peers in the swarm is smaller. On the other hand, fast peers can expect longer download times.

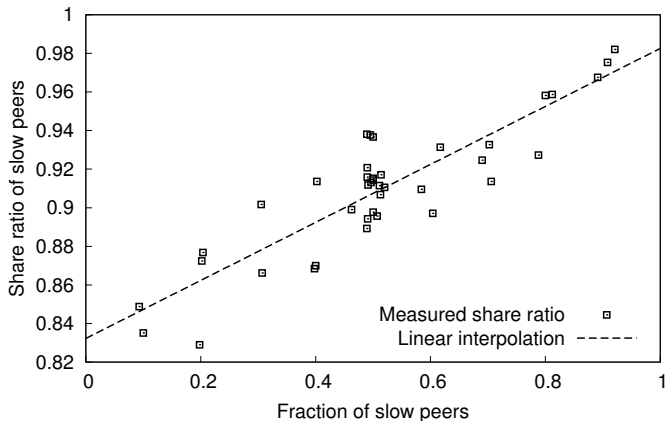


Fig. 5. Measured class share ratio against the proportion of slow peers for a 100-node swarm.

IV. ANALYTICAL MODELING

In this section, we present our analytical model for heterogeneous swarms. We model the utilization of available peer bandwidth according to the three phases we observed in our measurement study [2] and incorporate the clustering behavior among the peers with similar upload bandwidth. With this model, we can then estimate the download time for each class of peers in the system. We validate our model against PlanetLab experiments with more than 100 nodes.

The model of utilization of available peer bandwidth in Fig. 6 is the key to estimating the download time of each class. As discussed in Section III-A, ρ increases rapidly during the startup phase to a level ρ_0 , and stays there for the maximum utilization phase. Finally, it decreases in a step-like manner corresponding to the departure of peers from each class.

We can estimate the download time for each class of peers if we can accurately estimate the times taken by each step. To do so, we also consider the clustering phenomenon observed for BT nodes. In our model, we assume that the fastest peers unchoke only peers from the same class for the deterministic unchokes and that peers are picked at random for the optimistic unchokes and uniformly divided among the various classes of peers.

A. Model

We model a system with a flash crowd by assuming that it is a closed system consisting of a large number of peers, N , that arrive approximately at the same time. All peers attempt to download the same file, which is divided into M blocks of size B . The file is first made available at the seed which has an upload capacity C_s . Since upload capacity is limited in a closed system, we assume that the download bandwidth is unconstrained and that peers do not free-ride. Peers that download the file are divided into classes, with each class having $p_i N$ peers with c_i upload bandwidth. There are $r + 1$ classes with decreasing upload bandwidth (numbered from 0 to r). Class 0 is the fastest class of peers.

To estimate the download times expected for each class (T_i , for $i = 0, \dots, r$), we need to model the number of blocks that

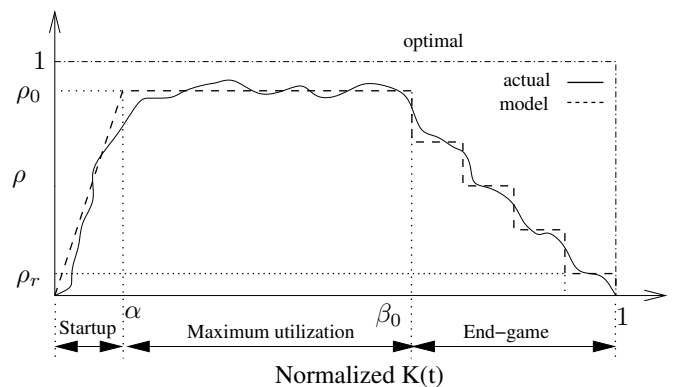


Fig. 6. Proposed model of ρ .

have been downloaded in the system by that time. We estimate K for each discrete time interval, Δt , using $\rho(t)$. In a closed system, assuming that the upload bandwidth of the server is fully utilized, the dynamics of the total number of blocks is given by:

$$K(t + \Delta t) = K(t) + \frac{C_s}{B} \Delta t + \frac{\rho(t) \sum_{i=0}^r p_i c_i}{B} \Delta t \quad (1)$$

where $\sum_{i=0}^r p_i c_i N$ is the total upload capacity of all peers in the system. We denote $\sum_{i=0}^r p_i c_i N$ with C .

As shown in Fig. 6, there are three distinct phases in our model: the startup phase from 0 to α , the maximum utilization phase from α to β_0 and the end-game phase from β_0 to 1. As observed from measurement, the end of a step corresponds to the fastest class finishing the download during the end-game phase. The parameters α , β_i and ρ_i , for $i = 0, \dots, r$, define the ρ curve. In this model, we assume the maximum utilization value (ρ_0) is one. While this is not entirely accurate in practice, our measurement results in Section III-A suggest that this approximation is good enough. We show next how to estimate these parameters and the download time for each class.

Where ρ_i is the value of ρ after all the peers in class i have left the system, we can model $\rho(t)$ as follows:

$$\rho(t) = \begin{cases} \rho_0 \frac{K(t)}{\alpha MN}, & K(t) \leq \alpha MN \\ \rho_0, & \alpha MN < K(t) \leq K(T_0) \\ \rho_i, & K(T_{i-1}) < K(t) \leq K(T_i), \\ & i = 1, \dots, r \end{cases} \quad (2)$$

We note that ρ_i decreases proportionally according to the upload bandwidth of the peers that have left the system as follows:

$$\rho_i = \begin{cases} \rho_0, & i = 0 \\ \rho_0 \left(1 - \frac{\sum_{j=0}^{i-1} p_j c_j}{C}\right), & i = 1, \dots, r \end{cases} \quad (3)$$

Using $\rho(t)$ in Equation (1) and solving using differential equations, we obtain:

$$K(t) = \begin{cases} \frac{C_s}{C} \frac{\alpha MN}{\rho_0} \left(e^{\frac{C}{B} \frac{\rho_0}{\alpha MN} t} - 1\right), & t \leq t_\alpha \\ \alpha MN + \frac{C_s + \rho_0 C}{B} (t - t_\alpha), & t_\alpha < t \leq T_0 \\ \frac{C_s + \rho_i C}{B} t + \varepsilon_i, & T_{i-1} < t \leq T_i, \\ & i = 1, \dots, r \end{cases} \quad (4)$$

where ε_i are the constants of integration. As $K(t)$ is a continuous function, the values for ε_i are derived as follows:

$$\varepsilon_i = \begin{cases} \alpha MN - \frac{C_s + \rho_0 C}{B} t_\alpha, & i = 0 \\ \sum_{j=1}^i \frac{(\rho_{j-1} - \rho_j) C}{B} T_{j-1} + \varepsilon_0, & i = 1, \dots, r \end{cases} \quad (5)$$

Using the $K(t)$ equation, we can derive t_α and T_i , for i from 0 to r :

$$t_\alpha = \frac{B}{C} \frac{\alpha MN}{\rho_0} \ln\left(\frac{C}{C_s} \rho_0 + 1\right) \quad (6)$$

$$T_i = (K(T_i) - \varepsilon_i) \frac{B}{C_s + \rho_i C}, i = 0, \dots, r \quad (7)$$

Because peers from class i finish the download and leave the system at T_i , Equation (7) gives us an estimate of T_i . However, this estimate is based on α and $K(T_i)$, which are still unknowns. Note also that in our derivations up to this point, we have not invoked the characteristics of the underlying BT protocol. It turns out that α and $K(T_i)$ are dependent on the actual p2p protocol employed in the system. For BitTorrent, we need to model α and $K(T_i)$ by taking into account observations about the choke/unchoke mechanism that regulates the trading of file blocks among the peers.

Choke and Unchoke in BitTorrent. In BT, peers start to upload data blocks to other peers only after they have received at least one piece, consisting of S blocks. Maximum utilization is achieved after all the peers have each downloaded at least one piece, initially from the server (seed), and are able to unchoke and upload to Q other peers. In this light, we use a conservative estimate and assume that maximum utilization is reached when the total number of blocks uploaded in the system is approximately NQS , where N is the number of peers in the system. This provides us with an estimate of α for BT:

$$NQS = \alpha MN \Rightarrow \alpha_{BT} = \frac{QS}{M} \quad (8)$$

This suggests that α is independent of the server bandwidth and the number of nodes. This is because BT has an optimistic unchoke mechanism (one optimistic unchoke out of a total of Q unchokes) that allows peers to download blocks from one another, and not only from the server.

Where i is the class of the fastest peers remaining in the system, we model the number of blocks downloaded in the system by time T_i . Due to clustering, we assume that the fastest class unchokes only peers from the same class (except the optimistic unchokes). Moreover, the fastest class takes up a fair share of the optimistic unchokes: $\frac{p_i}{\sum_{j=i}^r p_j}$ out of the total $\sum_{j=i}^r \frac{c_j}{Q} p_j N$ optimistic unchokes in the system. On the server side, we assume that it offers equal upload bandwidth to all peers in the system. This is consistent with our measurements. The time taken for the fastest class of peers to download $p_i MN$ blocks is:

$$\Delta t_i = \frac{p_i M N B}{\sum_{j=i}^r \frac{c_j}{Q} p_j N \frac{p_i}{r} + \frac{Q-1}{Q} p_i c_i N + C_s \frac{p_i}{r} \sum_{j=i}^r p_j} \quad (9)$$

During this time, peers in the other classes are downloading blocks. We estimate the number of blocks downloaded by the whole swarm by time T_i . The data downloaded by peers in other classes in Δt_i , denoted by $\Delta \kappa_i$:

$$\Delta \kappa_i = \Delta t_i \times \left(\sum_{j=i}^r \frac{c_j}{Q} p_j N \frac{\sum_{j=i+1}^r p_j}{\sum_{j=i}^r p_j} + \frac{Q-1}{Q} \sum_{j=i+1}^r p_j c_j N + C_s \frac{\sum_{j=i+1}^r p_j}{\sum_{j=i}^r p_j} \right) \quad (10)$$

Finally, we can compute $K(T_i)$, the total number of blocks downloaded in the system by time T_i .

$$K(T_i) = \begin{cases} \sum_{j=0}^i p_j MN + \frac{\Delta \kappa_i}{B}, & i = 0, \dots, r-1 \\ MN, & i = r \end{cases} \quad (11)$$

B. Validation

We validated the average download time of each class predicted by Equation (7) with experiments on PlanetLab. We ran experiments with more than 100 nodes with two, three and five classes of peers of different upload bandwidths. We ran 80 experiments with two classes of peers (64 and 128 kBps) with the number of nodes varying from 100 to 150, 20 experiments with three classes (64, 128, and 192 kBps) with 150 nodes and 16 experiments with 150 nodes divided among five classes (16, 32, 64, 128, and 192 kBps). The server bandwidth was fixed at 256 kBps. The proportion of peers in each class was varied to cover the range between 0 and 1.

Table I shows the errors we obtained when comparing the values from our model to the measured values from the experiments. We found that the errors for the two-class experiments are smaller than those for the three- and five-class experiments. On average, the errors for the two- and three-class experiments are less than 15% and this suggests that our model accurately predicts the expected download times for each class. The errors we obtained for the five-class experiments are higher because of the small number of peers in each class. The small difference

TABLE I
ERRORS IN ESTIMATING THE DOWNLOAD TIME.

No. of classes	Error (%) for each class (kBps)					Avg. error
	16	32	64	128	192	
2	-	-	9.6	11.9	-	10.7
3	-	-	15.8	5.6	18.1	13.1
5	26.4	33.6	12.1	26.4	30.5	29.5

in bandwidth among the classes also results in less precise clustering, which is a situation that is not fully captured in our model. Due to practical constraints on PlanetLab, we were not able to perform experiments with a larger number of peers and high upload bandwidth.

The inaccuracies in our estimates can be explained by real network conditions in which our experiments were run and the less than ideal share ratio described in Section III-B. Due to the network conditions, peers might experience delays that are not captured by our model. Moreover, we could improve the accuracy of our estimates by accounting for the smaller-than-one share ratio for the class of slow nodes.

Fig. 7 shows the impact of the proportion of slow peers on the download time of each class for a swarm with 100 nodes and a server bandwidth of 256 kBps. The lines represent the values predicted by our model, while the dots correspond to actual measurements. As shown by the dotted line, the download time for the slow peers (64 kBps) is affected more significantly than the download time of the fast peers (128 kBps) by the proportion of slow peers in the swarm. When the proportion of slow peers is increased, the download time for slow peers increases considerably, while that for the fast peers remains almost constant.

Our model seems to slightly overestimate the download time for the slow peers. The error is bigger for the fast peers, and our model slightly underestimates the average download time. This trend can be explained by the fact that the share ratio is slightly in the favor of the slow peers. The accuracy of the download time estimates for the fast peers when there is a large proportion of slow peers is lower because the clustering phenomenon is less pronounced. This leads to larger download times than those predicted by the model.

Fig. 8 shows the impact of the upload bandwidth of the slow peers on the download time of each class in a 100-node system with a 256 kBps server capacity and fast peers with 128 kBps upload capacity. Due to clustering, the download time of the fast peers is independent of the bandwidth of the slow peers. On the other hand, the download time of the slow peers sharply

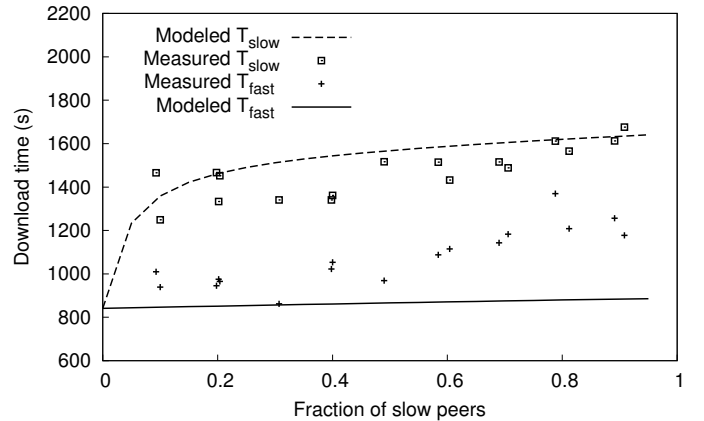


Fig. 7. Download time when varying the proportion of slow peers.

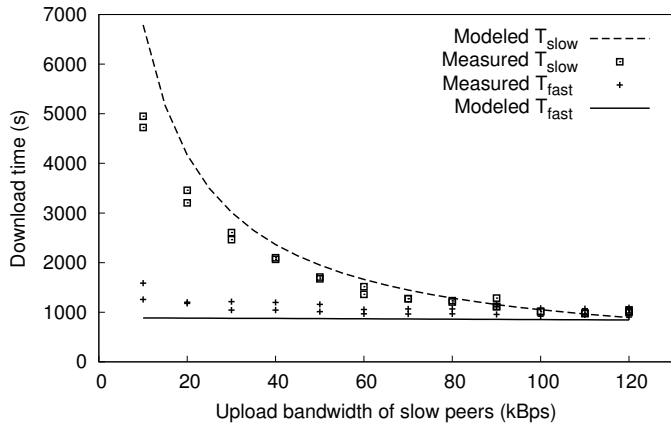


Fig. 8. Download time when varying the upload bandwidth of slow peers.

increases for small values of the upload bandwidth. This is expected, since the 50% of fast peers tend to cluster and finish faster, leaving just the slow peers to complete in a longer time. The dots represent the measured average download times for our experiments on PlanetLab. The error is larger for small values of the upload bandwidth of the slow peers. In this case, the slow peers benefit more when they get unchoked by the fast peers, even though they reciprocate the service and unchoke the fast peers. The investigation of the impact of bandwidth on the share ratio is left as future work.

V. APPLICATION: SERVER PROVISIONING

An important concern for file distribution systems is to offer sufficient server capacity so that clients achieve a minimal required quality of service. Unlike traditional client-server file distribution systems, the peers in the system will also contribute capacity, so the amount of server capacity required is not necessarily directly proportional to the number of supported clients. A content distributor needs to pay an ISP for the server bandwidth and it is costly to over-provision. Ideally, the server capacity allocated should be high enough to meet the quality of service requirements, and yet not excessive. Moreover, the unpredictability of flash crowds coupled with the heterogeneous bandwidth of peers also affects the required server capacity.

We show how our model can be used to find a server capacity that strikes a balance between maintenance costs and providing the required quality of service. While a closed form solution for the capacity of the server can be difficult to obtain for a heterogeneous system, we use our model to estimate the download time for different capacities of the server and plot the server capacity against download time. Assuming the existence of logs from previously served files with the estimated upload bandwidth of the peers and their distributions in different classes, we can plot this curve and estimate download times (quality of service) as the server capacity varies.

In Fig. 9, we plot the estimated server bandwidth needed for a specific download time for two swarms with 100 and 500 nodes.

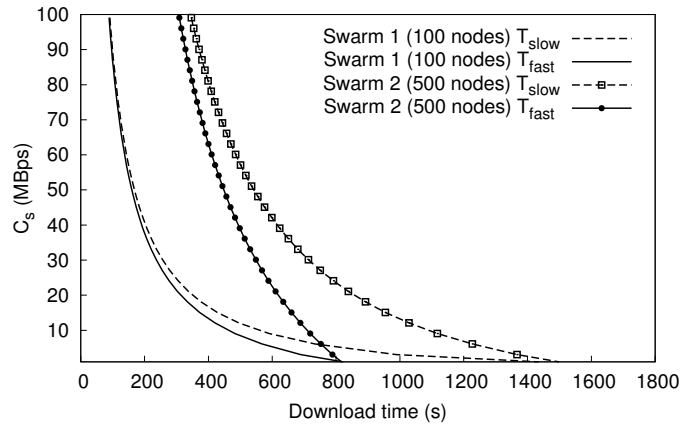


Fig. 9. Server bandwidth versus download time.

The nodes are equally divided into two classes with 64 kBps and 128 kBps upload capacity. We assume the quality of service requirements are expressed in terms of the maximum download time for each class of peers. It is unreasonable to expect all peers, regardless of their intrinsic upload bandwidth to finish at the same time. Hence, we can infer the server capacity needed for the slow class to finish in a specific time. For example, if the slow peers are expected to finish in less than 700 seconds, the server capacity needs to be at least 30 MBps for a system with 500 clients.

Fig. 9 also shows the impact of swarm size on the server capacity. The slow peers stay in the system longer than fast peers, hence they benefit more from the upload capacity of the server. Furthermore, Fig. 7 shows that our model overestimates the download time for slow peers and underestimates that for fast peers. Therefore, we expect that the actual values for the average download times of each class of peers will be situated between the slow and fast lines of each swarm in Fig. 9. Hence our model bounds the performance expected for the whole system.

In addition, we see that we need a considerable increase in server capacity to achieve a small improvement in the download time for the fast peers. This observation is especially important for large swarms, because an increase in server capacity hardly changes the download times for the fast peers, as shown in Fig. 9. For small swarms, increasing the server capacity can improve download times, but only up a certain point, i.e. 40 MBps in an 100-peer swarm.

Lastly, we can deduce from the model the server capacity required to achieve similar download times for all peers regardless of bandwidth. For example, our model suggests that the required server capacity is 90 MBps for the 500-node swarm and 40 MBps for the 100-node swarm. This analysis can be repeated with other system settings, such as server and peer upload bandwidth, and file size, among others.

VI. CONCLUSIONS

Our measurement study of BitTorrent on PlanetLab shows that the utilization of the available bandwidth of peers in a heterogeneous system has a fixed pattern over the course of a download and that peers enjoy an amount of service proportional with their contribution. Unlike homogeneous systems, the end-game phase for a heterogeneous swarm is a step-like function that corresponds to the number of peer classes. Moreover, although slower peers receive slightly more service than their upload contribution to the system, their class share ratio is close to that of a fair system.

Based on these insights, we developed an analytical model to estimate the download time for each class of peers by modeling the peer clustering during the end-game phase. We validated our model with experiments on PlanetLab and showed that it predicts the download time with an average error of 16.5%. Overall, our results suggest that BT is a good algorithm for peer-assisted file distribution and our model can be applied to estimate the server capacity required to achieve a desired quality of service. We observed that the server capacity has a higher impact on the download time of slow peers, and reducing the download time of fast peers in large swarms requires a significant increase in server capacity and cost.

As future work, we will extend our model to investigate the impact of the number of peer classes on class share ratio. The model can also be extended by adding download bandwidth constraints for peers. The accuracy of our model can be further improved by modeling the service distribution policies of the server, other incentive mechanisms, network connectivity and conditions, among others.

ACKNOWLEDGEMENT

This work was supported by the Singapore Ministry of Education grant R-252-000-348-112.

REFERENCES

- [1] A. R. Bharambe, C. Herley, and V. N. Padmanabhan. Analyzing and Improving a BitTorrent Networks Performance Mechanisms. In *Proc. of IEEE Conference on Computer Communications*, 2006.
- [2] C. Carbutaru, B. Leong, Y. M. Teo, and T. Ho. Modeling Transient Flash Crowd Performance for Peer-to-peer File Distribution Systems. Technical report, Department of Computer Science, National University of Singapore, July 2010.
- [3] N. Carlsson, D. L. Eager, and A. Mahanti. Using Torrent Inflation to Efficiently Serve the Long Tail in Peer-Assisted Content Delivery Systems. In *Proc. of IFIP Networking*, 2010.
- [4] A. L. H. Chow, L. Golubchik, and V. Misra. BitTorrent: An Extensible Heterogeneous Model. In *Proc. of IEEE Conference on Computer Communications*, pages 585–593, 2009.
- [5] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman. PlanetLab: An Overlay Testbed for Broad-Coverage Services. *Computer Communication Review*, 33:3–12, 2003.
- [6] F. Clevenot-Perronnin, P. Nain, and K. W. Ross. Multiclass P2P Networks: Static Resource Allocation for Service Differentiation and Bandwidth Diversity. *Performance Evaluation*, 62:32–49, 2005.
- [7] B. Cohen. Incentives Build Robustness in BitTorrent. In *Workshop on Economics of Peer-to-Peer Systems*, 2003.
- [8] G. Dán and N. Carlsson. Dynamic Swarm Management for Improved BitTorrent Performance. In *Proc. of International Conference on Peer-to-peer Systems*, 2009.
- [9] S. Das, S. Tewari, and L. Kleinrock. The Case for Servers in a Peer-to-Peer World. In *Proc. of IEEE International Conference on Communications*, 2006.
- [10] B. Fan, J. C. S. Lui, and D.-M. Chiu. The Design Trade-offs of BitTorrent-like File Sharing Protocols. *Transactions on Networking*, 2009.
- [11] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang. A Performance Study of BitTorrent-like Peer-to-peer Systems. *IEEE Journal on Selected Areas in Communications*, 25:155–169, 2007.
- [12] A. Legout, N. Liogkas, E. Kohler, and L. Zhang. Clustering and Sharing Incentives in BitTorrent Systems. *ACM SIGMETRICS Performance Evaluation Review*, 35:301–312, 2007.
- [13] Q. Li and J. C.-S. Lui. On Modeling Clustering Indexes of BT-Like Systems. In *Proc. of IEEE International Conference on Communications*, pages 1–6, 2009.
- [14] W. C. Liao, F. Papadopoulos, and K. Psounis. Performance Analysis of BitTorrent-like Systems with Heterogeneous Users. *Performance Evaluation*, 64:876–891, 2007.
- [15] D. S. Menasche, A. A. Rocha, B. Li, D. Towsley, and A. Venkataramani. Content Availability and Bundling in Swarming Systems. In *Proc. of International Conference on Emerging Networking Experiments and Technologies*, 2009.
- [16] M. Meulpolder, J. A. Pouwelse, D. H. J. Epema, and H. J. Sips. Modeling and Analysis of Bandwidth-inhomogeneous Swarms in BitTorrent. In *Proc. of Peer-to-Peer Computing*, pages 232–241, 2009.
- [17] V. Misra, P. Barford, and M. S. Squillante. Incentivizing Peer-assisted Services: A Fluid Shapley Value Approach. *ACM SIGMETRICS Performance Evaluation Review*, 2010.
- [18] R. S. Peterson and E. G. Sirer. Antfarm: Efficient Content Distribution with Managed Swarms. In *Proc. of USENIX Symposium on Networked Systems Design and Implementation*, pages 107–122, 2009.
- [19] M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy, and A. Venkataramani. Do Incentives Build Robustness in BitTorrent? In *Proc. of USENIX Symposium on Networked Systems Design and Implementation*, 2007.
- [20] D. Qiu and R. Srikant. Modeling and Performance Analysis of BitTorrent-like Peer-to-peer Networks. In *Proc. of ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pages 367 – 378, 2004.
- [21] F. Simatos, P. Robert, and F. Guillemin. A Queueing System for Modeling a File Sharing Principle. *ACM SIGMETRICS Performance Evaluation Review*, 36:181–192, 2008.
- [22] Y. Sun, F. Liu, B. Li, and B. Li. Peer-assisted Online Storage and Distribution: Modeling and Server Strategies. In *Proc. of International Workshop on Network and Operating Systems Support for Digital Audio and Video*, 2009.
- [23] Y. Tian, D. Wu, and K. Ng. Modeling, Analysis and Improvement for BitTorrent-like File Sharing Networks. In *Proc. of IEEE Conference on Computer Communications*, pages 1–11, 2006.
- [24] C. Wu, B. Li, and S. Zhao. Multi-Channel Live P2P Streaming: Refocusing on Servers. In *Proc. of IEEE Conference on Computer Communications*, 2008.
- [25] X. Yang and G. Veciana. Performance of Peer-to-peer Networks: Service Capacity and Role of Resource Sharing Policies. *Performance Evaluation, P2P Computing Systems*, 63:175 – 194, 2006.