Modeling Flash Crowd Performance in Peer-to-peer File Distribution

Cristina Carbunaru, Yong Meng Teo, Ben Leong, and Tracey Ho

Abstract—Given the growing popularity of peer-to-peer file distribution in commercial applications, it is important to understand the challenges of using p2p file-sharing protocols for file distribution, and how extreme conditions such as flash crowds affect the efficiency of file distribution. In this light, there is a need to understand the impact of the utilization of available bandwidth on the performance of peer-assisted file distribution systems. With a simple measurement study on PlanetLab, we identified distinct phases in peer bandwidth utilization over the download duration. Based on the evolution of the utilization of available peer bandwidth over time, we formulated an analytical model for flash crowds in homogeneous and heterogeneous bandwidth swarms. The model estimates the instantaneous download rate and the average file download time with 10% error for swarms up to 160 peers. Our model can be used to predict the scalability of the system when the number of peers increases, and to provision for flash crowds by estimating the server bandwidth to achieve a minimum quality of service. Lastly, we demonstrate how our model is applied to new p2p protocols to understand their design and performance problems.

Index Terms—peer-to-peer analytical modeling, performance model

1 Introduction

With the increase in the size of downloaded files over the Internet, peer-to-peer (p2p) file-sharing protocols such as BitTorrent (BT) have been widely adopted to improve the performance of traditional client-server file distribution systems [2], [10], [34]. Peer-assisted systems improve the scalability and performance of content distribution by utilizing 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 efficient in utilizing the available bandwidth of the system [26]. While new protocols [25], [30] have been proposed to address this issue, the popularity of BT and the availability of many implementations make it an attractive choice for file distribution in practice [2], [34]. In this light, there is a need to model and understand the performance of peer-assisted file distribution systems.

Previous work on modeling p2p systems focused mainly on systems at *steady-state* [12], [28], [33]. In file-sharing systems, modeling steady-state is reasonable because peers stay in the system and continue to share the file after download completion. In contrast, peers in file distribution systems download the file as fast as possible and then leave. Furthermore, when a popular file is made available, there is typically a surge in peer arrival rate which results in a *flash crowd* [36], as shown in Fig. 1. Subsequently, as the file popularity drops, peer arrival rate decreases and the system goes into steady-state. For a content distributor, the challenge is to ensure that the system has sufficient resources to cope with this sudden surge of users.

In this paper, we investigate flash crowd performance in peer-assisted file distribution systems. First, measurement experiments of BT on PlanetLab [4] show that the *utilization of available peer bandwidth* (ρ) is not constant over the download interval. Second, we propose an analytical model to capture the impact of the bandwidth utilization on the instantaneous download rate experienced by peers. Lastly, we show how users, service providers and developers of peer-assisted file distribution systems can apply our model to predict the expected download performance of the system for existing and new protocols, and to estimate the server bandwidth for achieving a minimum quality of service.

An important measurement observation is that peers are not able to fully utilize their entire available upload capacity during a flash crowd. By observing the variation of ρ as a function of the total number of blocks downloaded, both the file block availability and download performance over time are captured. For current p2p protocols, we show that ρ can be characterized by a trapezoidal-shape curve with three phases: *start-up*, *maximum utilization* and *end-game*.

Based on these measurement observations, we propose a new approach for modeling and predicting the instantaneous flash crowd performance of p2p file distribution systems. Our approach is designed for both homogeneous and heterogeneous bandwidth

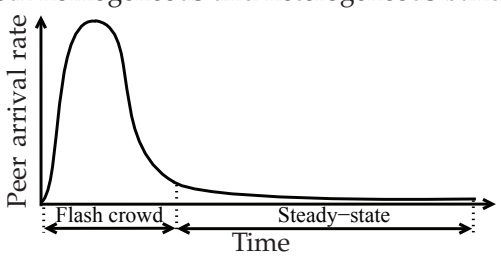


Fig. 1: Evolution of the peers arrival rate in swarms.

Digital Object Indentifier 10.1109/TPDS.2013.220

swarms. A key strength of our approach is that we can model the evolution of ρ in a file distribution system as an analytical function of the total number of blocks downloaded by all peers in the system. To simulate the flash crowd scenario, we model the swarm as a closed system. By transforming this model into the time domain, we estimated the system performance with high accuracy. A closed form solution is obtained to predict the file download rate (throughput) variation over time and the expected average download time experienced by the peers.

We apply our model to BT [5], a well-known and frequently used p2p protocol for file distribution. Validation using PlanetLab shows that the measured utilization of available peer bandwidth closely follows our model. While existing models [28], [35] only estimate the average download rate of the system, our model is designed to estimate the evolution of peer download rate over time. We validate our model with experiments on PlanetLab and show that, on average, it estimates the instantaneous download rate and the file download time with less than 10% error for swarms up to 160 peers. Our heterogeneous model is more accurate when compared to a simplistic model.

Next we apply our model to study the performance of p2p file distribution with flash crowd. First, our analytical approach provides a reasonably tight upper bound on the achieved download performance as the number of peers scales up. Second, we show how the server capacity required to support a specified quality of service is derived. This can help p2p service providers handle flash crowds without providing excessive bandwidth. Third, protocol designers can apply our approach to understand the performance of different p2p protocols. As an example, we applied our model to FairTorrent [30] with a different incentive scheme, and to BT swarms with peers that seed after download completion. We observed that FairTorrent suffers from starvation, especially when the system is under-provisioned or when the number of peers is large.

2 Related Work

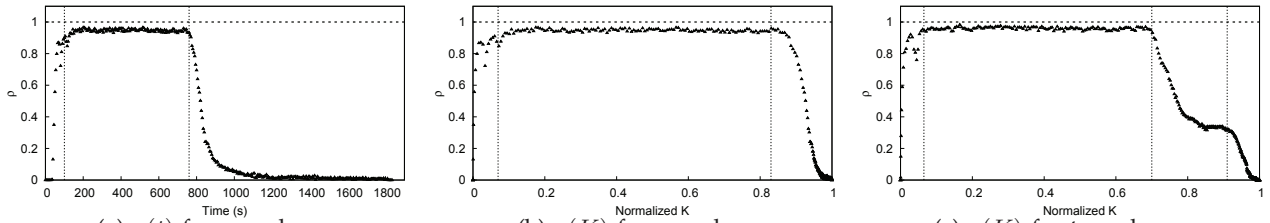
Unlike previous work that proposed new centrally coordinated mechanisms [25] and new pricing mechanisms to incentivize uncoordinated p2p schemes [23], we study the effectiveness of using the popular BT algorithm for file distribution. While the performance of BT has been studied extensively as a file-sharing protocol [9], [28], [31], to the best of our knowledge, we are the first to study the performance of BT as a file distribution protocol. In file distribution, the system provides a *server (BT seed)* as a constant source of data content for clients that download a file and clients generally leave the system on download completion.

There is a large body of work on the modeling of p2p systems. The most common approaches are queueing theory [11], [31], [33], [35] and fluid models [12], [17], [19], [20], [22], [28]. The main drawback of queueing models is that sufficiently detailed models are often mathematically intractable, while simple models fail to provide much insight. On the other hand, fluid models are useful because they capture the evolution of p2p systems over time. However, current work includes the assumption of constant arrival rate and models are solved for systems at steady-state [20], [28]. In terms of validation, simulation is widely used in previous models [9], [11], [19], [20], [31]. Typically, only parts of the models are validated using traces from real systems, while simulation is used to complete the validation [12], [28], [35]. Considering the unpredictability of flash crowds, we validated our model with PlanetLab measurements.

In reality, the worst case performance of p2p systems is often the result of flash crowd [13], [14], [35], [36]. While Yang and Veciana studied the ramp-up service capacity [35], a closed form solution for the expected download time is not provided, and the insight is that expected peer delay during the initial transient phase scales logarithmically with swarm size. In contrast, our closed-form model captures the lifetime performance of flash crowd.

The measure of effectiveness of file-sharing, η, is a key input parameter in the analytical models proposed by Yang and Veciana [35] and Qiu and Srikant [28]. The rationales are that the downloading peer's contribution to the service capacity is a fraction η of that of a peer that has fully downloaded the file, and the total capacity of the peers is fully utilized at steady-state. However, our measurement of the effective peer upload bandwidth utilization, ρ, shows that this assumption does not necessarily hold in practice, in particular during flash crowd. In reality, p2p protocols have different ρ when distributing the same file under similar network conditions [18].

Applications of previous models include estimates of minimum download time [15], analysis of free-riding [19], [28], [35] and swarm lifetime [12], [21], and scalability of video on-demand system during flash crowd [6]. In this paper, we focus on performance scalability of file distribution and server provisioning during flash crowds. On server provisioning, methods such as bandwidth allocation among peers [3], [7], content bundling [21], and dynamic allocation of peers among swarms [7], have been proposed to improve the download time and availability in p2p systems. The impact of server capacity on the performance of homogeneous peer-assisted systems has been studied using fluid models [8], [32], but we also cover heterogeneous model. Other studies [29] attempt to estimate the server provisioning for flash crowds, but only the last phase of exponential decay of the arrival rate [12] is modeled. A key difference is that we analyze the impact of multiple classes of peers on the required server capacity to achieve a specific download time during flash crowd.



(a) $\rho(t)$ for one class. (b) $\rho(K)$ for one class. (c) $\rho(K)$ for two classes.

Fig. 2: ρ for BT homogeneous and heterogeneous 100-node swarms.

3 Utilization of Available Peer Bandwidth

To investigate the performance of BitTorrent [5] as a file distribution protocol, we conducted measurement experiments on PlanetLab [4]. A key observation is that there is a consistent pattern in the utilization of the available bandwidth over the course of a download. In this section, we present the background information on existing p2p systems, explain our experimentation methodology and describe our observations on the utilization of available peer bandwidth.

In BT, peers in a swarm cooperate to download large files, initially only available on a few nodes that are called *seeds*. Peers simultaneously download and upload different parts of the file from other peers, as well as directly from the seeds. 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 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 remaining requests the peer received.

In our measurement study on PlanetLab, each experiment involves a tracker, a client that acts as the server or initial seed (which remains in the system throughout the experiment) and clients that act as peers. To mimic a file distribution scenario where the clients are only interested in downloading a file and not in helping others with their downloads, peers join the system at approximately the same time. We show in Appendix A (Supplementary Material) that this scenario allows us to obtain a good approximation of the performance of the peers arriving during a practical flash crowd scenario, where more peers continue to join the swarm at a much lower rate after the initial flash crowd. The intuition for this is that while the peers joining after a flash crowd leach some of the bandwidth of the initial peers, they also contribute some bandwidth to the swarm.

Since the upload capacity of nodes on PlanetLab is unknown, we cap the upload bandwidth 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 160 nodes and a maximum upload bandwidth of 256 kBps.

3.1 Definition and Observations

Previous analysis of the effectiveness of BT showed that available bandwidth can be approximated as one at steady-state [28]. An interesting question is what is the utilization of peer bandwidth during flash crowd. We define the utilization of available peer bandwidth as follows.

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

Based on more than 300 experiments with different configurations on PlanetLab, we observed that the evolution of bandwidth utilization during flash crowd can consistently be divided into three main phases: *start-up*, *maximum utilization* and *end-game*. For illustration, we plot in Fig. 2a the utilization for a homogeneous system where the server has an upload capacity of 256 kBps and all peers have an upload capacity of 128 kBps. In this example, the start-up phase is from 0 s to 100 s; the maximum utilization phase is from 100 s to 770 s, and a end-game phase is from 770 s to 2100 s. The maximum utilization phase resembles the steady-state with a constant value of η over time as reported in [28]. However, the maximum utilization phase of the flash crowd is followed by a steep decrease in ρ that precedes the steady-state.

To better understand the utilization of the available system bandwidth as 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. 2b. 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 postulate, and subsequently verify experimentally, that K captures the salient features of the evolution of the system. 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 MN.

3.2 The End-game Phase

A key difference between homogeneous and heterogeneous systems lies in the end-game phase. 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 phase contains *steps* that correspond with the classes of peers and steps occurr with the departure of the fastest peer in that class. In Fig. 2c, we plot ρ against $K(t)$ for swarms with heterogeneous peers equally divided in two classes. The upload bandwidths of slow peers, fast peers and server are 64 kBps, 128 kBps and 256 kBps, respectively. As shown, the steps occurrence in the end-game phase are at 0.7 and 0.9, respectively. This observation is consistent for systems with a larger number classes.

These steps concur with the observation that peers tend to cluster with peers of similar upload bandwidths, as highlighted by Legout et al. [16]. In the ideal case, when clustering is perfect, peers in a class finish their downloads close in time. However, in practice, the upload capacities of the peers are influenced by network conditions and connectivity, hence the steps in the end-game phase are less distinct. When the number of classes increases, step size reduces, and the profile of ρ resembles a homogeneous system.

4 Modeling A Flash Crowd

In this section, we describe our approach in modeling the impact of the utilization of available peer bandwidth (ρ) during a flash crowd for p2p file distribution systems. For clarity, we first focus on modeling a homogeneous bandwidth system. We extend this model to heterogeneous swarms with peers divided among different classes in Section 4.3.

The utilization of available peer bandwidth is the key factor used in our model. Based on measurement experiments, the three phases for a file download *start-up*, *maximum utilization* and *end-game* can be approximated with a trapezoidal-shape curve and modeled as shown in Fig. 3. If the upload capacities of all peers can be fully utilized at all times ($\rho = 1$), optimal performance can be achieved. This does not happen in practice, as described in Section 3.1. By modeling ρ and the parameters characterizing these phases, α, β, ρ_{max} and ρ_{min}, average download time and download rate variation over time can be derived. Our model has a closed form solution and can be used for different p2p protocols.

4.1 General Model

This section describes a general modeling approach for predicting download rate variation over time and download times. It has been shown that the access patterns of popular and newly available content typically experience what is called a flash crowd [14], [27]. We model a flash crowd as a closed system consisting of a large number of peers, N, arriving at approximately the same time. In Appendix A, we show that this impulse-like arrival pattern is a good approximation of the performance observed in a typical flash crowd scenario where more peers continue to join the swarm, but albeit at a much lower rate of arrival than the initial flash crowd. All peers attempt to download the same file, which is divided into M blocks of size B. The file is first made available by a peer, called a seed (server), with an upload bandwidth C_s. Peers that download the file have maximum upload bandwidth c_i, for $i = 1, \cdots, N$ and leave as soon as the file is downloaded. The notations are summarized in Table 1.

To determine the download rate and time, we first estimate $K(t)$, the total number of blocks downloaded in the system by time t. The evolution of K is modeled over discrete time intervals, Δt, where Δt is arbitrarily small. $K(t)$ depends on the utilization of available peer bandwidth at time t, denoted by $\rho(t)$. The total number of downloaded blocks increases due to server and peers' contribution. We assume that the server's upload capacity is fully utilized, while peers might not use their maximum upload capacity all the time. Hence only a fraction ρ of the upload capacity is used at t. The evolution of the total number of blocks over time is estimated as follows:

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

where $\sum_{i=1}^{N} c_i$ is the total upload capacity of the peers in the system. For the rest of the paper, we denote $\sum_{i=1}^{N} c_i$ with C.

4.2 Homogeneous Model

The evolution of ρ for a homogeneous system is modeled as three distinct phases as shown in Fig. 3. During *start-up*, the peers join the system and the server is the only one offering file blocks. It takes some time before the peers accumulate enough blocks to start exchanging them. As the peers download their first blocks from the server, the utilization increases, reaching the full capacity of the system. When the peers download a fraction α of the total number of blocks needed by all peers to complete the download, the utilization reaches a maximum value ρ_{max}. This *maximum utilization* phase continues until the moment

TABLE 1: Model notation.

Notation	Description
N	number of peers in the system
M	number of blocks in the file
B	size of a block
S	number of blocks in a BT piece
Q	number of simultaneous unchokes allowed in BT
C_s	maximum upload bandwidth of the server
c_i	maximum upload bandwidth of peer i
$\rho(t)$	utilization of available peer bandwidth at time t
α	fraction of blocks when maximum utilization is reached
β	fraction of blocks when utilization starts to decrease
ρ_{max}	maximum utilization of available peer bandwidth
ρ_{min}	minimum utilization of available peer bandwidth
$K(t)$	total number of blocks downloaded in the system by time t
$r_d(t)$	download rate at time t
T_d	average download time

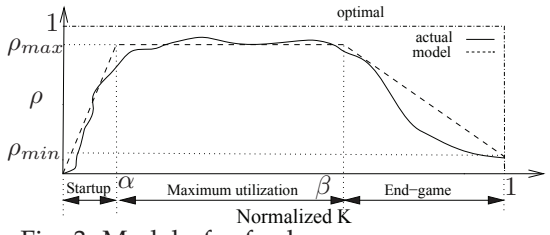


Fig. 3: Model of ρ for homogeneous swarm.

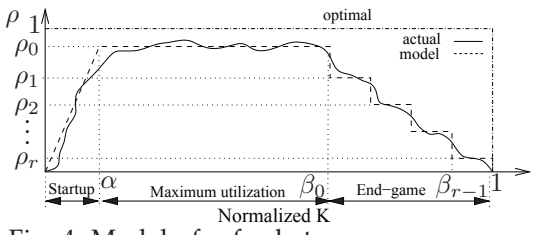


Fig. 4: Model of ρ for heterogeneous swarm.

when the first peer completely downloads the file, and this marks the start of the *end-game* phase. At this point, a total of βMN blocks would be downloaded by the nodes and the utilization decreases to reach ρ_{min} at the end of the download. ρ_{min} is 0 when there is no altruistic sharing, and it is greater than 0 otherwise.

Each phase of $\rho(K)$ is modeled using a linear function. As shown in Fig. 3, $\rho(K)$ is fully described by the four parameters: α, β, ρ_{max} and ρ_{min} as follows:

$$\rho(K) = \begin{cases} \rho_{max}\frac{K(t)}{\alpha MN}, & K(t) \leq \alpha MN \\ \rho_{max}, & \alpha MN < K(t) \leq \beta MN \\ \frac{(\rho_{max}-\beta\rho_{min})-\frac{K(t)}{MN}(\rho_{max}-\rho_{min})}{1-\beta}, & \beta MN < K(t) \leq MN \end{cases} \quad (2)$$

Substituting $\rho(K)$ in Equation (1) and solving this using differential equations, we obtain a closed form solution for $K(t)$. Furthermore, download rate over time ($r_d(t)$) and average download time (T_d) are derived from $K(t)$:

$$r_d(t) = \begin{cases} \frac{C_s}{N}e^{\frac{C}{B}\frac{\rho_{max}}{\alpha MN}t}, & t \leq t_\alpha \\ \frac{C_s+\rho_{max}C}{N}, & t_\alpha < t \leq t_\beta \\ \frac{C_s+\rho_{max}C}{N}e^{-\frac{C}{B}\frac{\rho_{max}-\rho_{min}}{(1-\beta)MN}(t-t_\beta)}, & t_\beta < t \end{cases} \quad (3)$$

$$T_d = t_\beta - \frac{B}{C}\frac{(1-\beta)MN}{\rho_{max}-\rho_{min}}\ln\frac{C_s+\rho_{min}C}{C_s+\rho_{max}C}, \quad (4)$$

where

$$t_\alpha = \frac{B}{C}\frac{\alpha MN}{\rho_{max}}\ln(\frac{C}{C_s}\rho_{max}+1) \quad (5)$$

$$t_\beta = (\beta-\alpha)\frac{BMN}{C_s+\rho_{max}C} + t_\alpha \quad (6)$$

Details of the model derivation is in Appendix B.1.

We note that our model is independent of the chosen p2p protocol, but in modeling ρ, we need to consider the protocol characteristics. Appendix B.1.2 describes how we model the parameters that define the utilization of available bandwidth: α, β, ρ_{max}, ρ_{min}. Parameter ρ_{max} is estimated using measurement, $\rho_{min} = 0$ for BT, and other parameters are estimated as follows:

$$\alpha = \frac{QS}{2M} \quad (7)$$

$$\beta = \alpha + (C_s+\rho_{max}C)\frac{M-SQ}{M(C_s+(1+f)\rho_{max}C)} \quad (8)$$

4.3 Heterogeneous Model

Real p2p systems are typically heterogeneous. In this section, we extend our model to heterogeneous systems, with several classes of peers, where the peers within each class have the same upload bandwidth. We believe that this is a reasonable assumption because ISPs commonly sells a limited number of subscriber plans.

As discussed in Section 3.2, the major difference between homogeneous and heterogeneous systems is in the end-game phase. The model of utilization of available peer bandwidth for a heterogeneous system is shown in Fig. 4. Based on ρ, we estimate the download time for each class of peers by accurately estimating the times taken by each step. To do so, we also consider the clustering phenomenon observed for BT nodes. For deterministic unchokes, we assume that the fastest peers unchoke only peers from the same class. In optimistic unchokes, peers are picked randomly and unchokes are uniformly divided among the classes of peers. $\rho(K)$ is modeled as:

$$\rho(K) = \begin{cases} \rho_0\frac{K(t)}{\alpha MN}, & K(t) \leq \alpha MN \\ \rho_0, & \alpha MN < K(t) \leq K(T_0) \\ \rho_i, & K(T_{d_{i-1}}) < K(t) \leq K(T_{d_i}), \\ & i = 1, \cdots, r \end{cases} \quad (9)$$

$$\text{where } \rho_i = \begin{cases} \rho_0, & i = 0 \\ \sum_{j=0}^{i-1} p_j c_j & \\ \rho_0(1-\frac{\sum_{j=0}^{i-1} p_j c_j}{C}), & i = 1, \cdots, r \end{cases} \quad (10)$$

Similar to the homogeneous model, the download time expected for each class i of peers is:

$$T_{d_i} = (K(T_{d_i}) - \varepsilon_i)\frac{B}{C_s+\rho_i C}, i = 0, \cdots, r \quad (11)$$

Details can be found in Appendix B.2.

5 Validation

The model is validated against measurements conducted on PlanetLab [4]. Using the Python BT implementation [1], we modified the client program to quit after completing the download. The file size of 100 MB is divided into blocks of 16 kB with 16 blocks forming a piece. To validate the homogeneous model, we ran 170 experiments with the number of peers varying between 20 and 160 nodes. We set the upload peer bandwidth to 128 kBps and the server bandwidth varied between 128 kBps and 4 MBps. For the heterogeneous model, we also ran 120 experiments, each with 40 to 150 nodes divided among two classes (64 and 128 kBps), three classes (64, 128, and 192 kBps), and five classes (16, 32, 64, 128, and 192 kBps). We also varied the percentage of peers in each class.

Peers in a swarm are simultaneously started on all PlanetLab machines to mimic flash crowd. Peer actions are recorded in a log file with the event time and details of file blocks downloaded. These events are processed at discrete time intervals of five seconds. The interval is sufficiently large to observe variations in the system, and small enough to estimate the instantaneous download rate.

From measurement observation, ρ_{max} is protocol-dependent. However, for a specific protocol, it is approximately constant for different swarm sizes. Analysis of BT utilization profile shows that 75% of the values of ρ_{max} fall between 0.89 and 0.95. Hence, we use a value of 0.93 for all our experiments. Further details on model validation and comparison with a simplistic model are in Appendix C.

5.1 Download Rate and Download Time

To validate the homogeneous and heterogeneous models, we compare the measured values with the estimated values in Equations (3), (4) and (11), respectively.

For the homogeneous download rate in Equation (3), we plotted for each experiment the measured values and the values obtained from our model. In all our experiments, we found that the model closely follows the measured download rate fluctuation over time. In Fig. 5, we show one instance of a BT experiment with 100 nodes. On average for all experiments, the modeled download rate over time differs from the measured values by 7.3%, and, in more than 80%, the estimates differ by less than 10%. In summary, the average download time difference between Equation (4) and measurement is 6.7% for the 170 BT experiments conducted with 20 to 160 homogeneous peers. For each pair of analytical and measured results, relative errors are maximum 20% and decrease with a larger number of peers. We conclude that our homogeneous model estimates the average download time with good accuracy.

We also validated Equation (11) in the heterogeneous model against the measured average download time for each class of peers. Table 2 shows the relative errors for swarms up to 160 peers and, as expected, the errors tend to increase with more classes of peers.

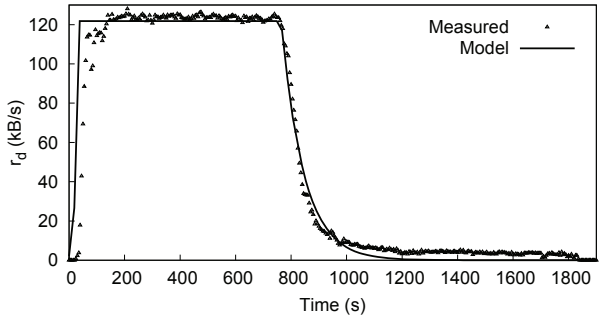


Fig. 5: Validation of $r_d(t)$ for 100-node BT homogeneous swarm ($c_i = 128$ kBps, $C_s = 256$ kBps).

TABLE 2: Errors in estimating the download time for the homogeneous and heterogeneous models.

No. of classes	Error (%) for each class (kBps)				
	16	32	64	128	192
1	-	-	-	6.7	-
2	-	-	9.6	11.9	-
3	-	-	15.8	5.6	18.1
5	26.4	33.6	12.1	26.4	30.5

We see that the errors for the slower and faster classes are larger than for the medium-bandwidth classes. Moreover, we consistently observed that our model overestimates the download time for slow peers and underestimates that for fast peers. We further discuss this observation in Section 6.1. One cause of errors is likely due to practical network conditions in our experiments, which are not captured in our model. For example, peers might experience network delays that are not modeled. Moreover, our model of the "step-like" nature of the end-game phase becomes less accurate with more classes of peers and when the bandwidth difference among classes is small.

We may not always need an estimate of the average download times for each class of peers. If we only care about the average download time of the system, instead of attempting to apply the heterogeneous model, another possible approach is to apply the homogeneous model with the average upload bandwidth of the peers as an input to the model. In Table 3, we compare the errors in estimating the average download time of the whole swarm when using these two approaches. As expected, the heterogeneous model provides better estimates, but the gap reduces with more peer classes. In particular, for the five-class swarms, we found that the estimated average download time of the system using the heterogeneous model is 18% away from the average download time obtained through measurements. This error is smaller than that for the estimates for various classes shown in Table 2, except the 64 kBps class. The corresponding error for the estimate with the homogeneous model is 22.3%. This suggests that for swarms with a large number of peer classes, the homogeneous model is good enough when only the system average download time is needed.

5.2 Model Sensitivity to Parameter Variation

This section investigates the sensitivity of our model to errors in the parameter estimates. To do so, we

TABLE 3: Errors in estimating the download time using homogeneous model for heterogeneous swarms.

No. of classes in swarm	Error (%)	
	Heterogeneous model	Homogeneous model
2	7.1	13.4
3	8.1	16.7
5	18.0	22.3

TABLE 4: Sensitivity analysis of proposed model.

Parameter - x	Absolute Variation dx	Download Time Variation (%) $\frac{dT_d}{T_d}$
$0.01 < \alpha < 0.03$	0.01	2.1
$0.71 < \beta < 0.93$	0.10	21.4
$0.83 < \rho_{max} < 0.97$	0.10	8.9

quantify the impact of changes in the parameters, α, β, and ρ_{max}, on the estimates of the download time. We do this by differentiating the download time, T_d, with respect to the various parameters in Equation (4), i.e. $\frac{dT_d}{d\alpha}$, $\frac{dT_d}{d\beta}$, and $\frac{dT_d}{d\rho_{max}}$. We then compute dT_d for a given variation of the parameters values that is chosen within the range of reasonable parameter values.

Table 4 shows our results for BT. First, we observe that the download time is the most sensitive to β and ρ_{max}, while α has little impact on the final result. An absolute difference of 0.10 in estimating β and ρ_{max} results in an expected download time variation of 21% and 9%, respectively. This suggests that β can have a large impact on the final estimate of the model. However, because the measured values for β lie between 0.71 and 0.93 (or a range of 0.22), the error in the download time estimates arising from an error in β is likely to be within 20%. Similarly, the error in the download time estimate arising from an error in estimating ρ_{max} is likely to be no more than 13%. On the other hand, an absolute variation of 0.01 for α impacts only the start-up phase of the download, and thus has a negligible effect on the estimated download time.

6 Model Applications

In this section, we describe three applications of our p2p model. First, p2p users can apply our model to determine the expected download time during flash crowds. Next, service providers can use our model to determine the required server capacity to achieve a specific quality of service in p2p systems. Finally, p2p protocol designers can use the model for the utilization of available peer bandwidth to evaluate the design trade-offs analysis for new protocols.

6.1 Scalability

Using our model, we estimate and analyze how the download time varies for p2p systems configurations that are larger than what we can measure on PlanetLab. Next we present the scalability analysis for heterogeneous systems. The analysis for homogeneous systems is in Appendix D.1.

Fig. 6 shows the variation of download time for each class in a swarm with two peer classes and with increasing number of peers. The swarm contains 50% slow peers with 64 kBps upload bandwidth and 50% fast peers with 128 kBps upload bandwidth and a server capacity of 256 kBps. Lines in this figure represent the estimated values and the dots represent the measured values on PlanetLab. Like the homogeneous case, we observe that the download times expected by each class scales well when the number of peers in the swarm is increased. Fig. 6 also shows that the influence of the server capacity is more pronounced for the slow peers than for the fast peers. Hence the download time of the slow peers stabilizes when the swarm size is greater than 60, while that of the fast peers stabilizes for swarms smaller than 40 peers. The measured values follow the trend given by the analytical results, even though there are small errors in the estimates.

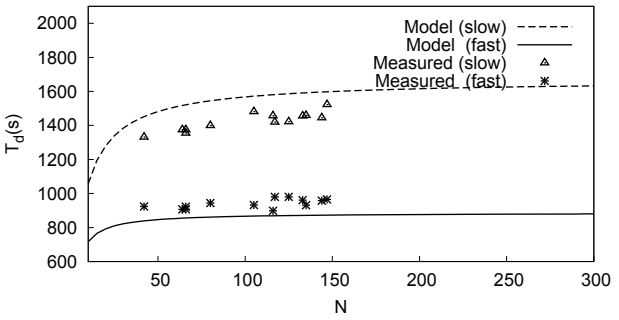


Fig. 6: Scalability of BT swarms with two classes.

Furthermore, Fig. 6 shows that our model underestimates the download time for the fast peers and slightly overestimates the download time for the slow peers. This trend, where the actual measured values for the average download times of each class is bounded by the estimates for the slowest and fastest class, is consistent for swarms with different number of classes.

There are two reasons for underestimating the download time of the fastest peers. First, we tend to overestimate the download rate of the fastest peer because we assume that it downloads only from the fastest peers. Second, the step function of our model implicitly assumes that all the peers of a class leave the swarm at the same time. In practice, some peers leave later than others. Because our model predicts that the faster peers leave the system earlier than what we observe in practice, our model underestimates the total upload capacity left in the system after the peers start leaving the system. This results in overestimating the download time of the slowest class of peers left in the system.

6.2 Server Bandwidth Provisioning

An important concern for file distribution systems is to offer sufficient server capacity so that clients achieve at least a minimal required quality of service. Unlike traditional client-server file distribution systems, peers in the system will also contribute capacity, so the amount of server capacity required is not directly proportional to the number of supported clients. A content distributor needs to pay for server bandwidth, thus it is costly to over-provision. Ideally, the server capacity allocated should be high enough to

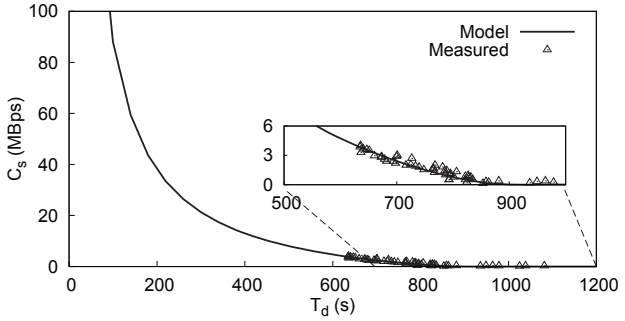


Fig. 7: Server capacity for 100-node BT homogeneous swarm - varying the download time.

meet the quality of service requirements, and yet not excessive. With our model, we can predict the server capacity for both homogeneous and heterogeneous systems.

6.2.1 Provisioning for Homogeneous Systems

In this section, we analyze the impact of the utilization of available peer bandwidth on the server capacity needed in a file distribution swarm. Using our model, we derive the required upload server capacity to achieve a required download time in a homogeneous swarm. Details of the derivation are in Appendix B.1.1.

To understand the impact of changing the requirements for download time on server capacity, we show in Fig. 7 the estimates for a BT swarm with 100 nodes and a peer capacity of 128 kBps. The lines correspond to the predictions obtained from Equation (37), while the points are actual PlanetLab measurements. We observe that the required server capacity increases exponentially when the download time is decreased. When the required download time is less than 600 seconds, the server becomes the major provider in the system. Beyond this point, the contribution of the peers has little impact on performance. As a consequence, slightly relaxing the download time by 10%, from 500 seconds to 600 seconds, may lead to a 40% decrease in the required server capacity. In contrast, if we do not require a download time that is faster than 600 seconds, the contribution of the peers has a much more significant impact on the download time. Therefore, our model can help content providers understand the trade-off between download time requirements and provisioning costs.

6.2.2 Provisioning for Heterogeneous Systems

The unpredictability of flash crowds coupled with the heterogeneous bandwidth of peers affects the required server capacity. While a closed form solution for the capacity of the server can be difficult to obtain for a heterogeneous system, we can use our model to estimate the download time for different server 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

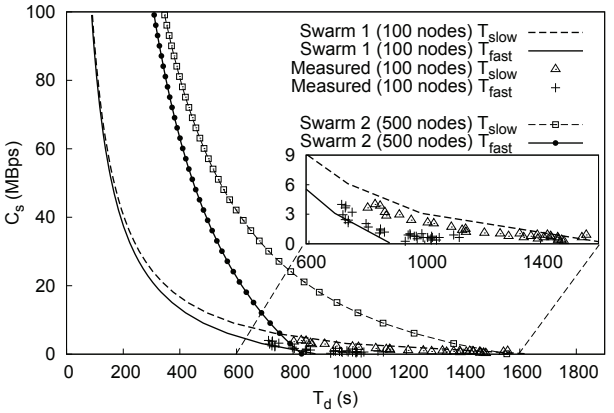


Fig. 8: Server capacity for BT heterogeneous swarms - varying the download time.

in different classes, we can plot this curve and estimate download times (quality of service) as the server capacity varies.

In Fig. 8, we plot using lines the estimated server bandwidth needed for a specific download time for two swarms with 100 and 500 nodes. 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.

Furthermore, Fig. 8 shows the measured average download times for swarms with 100 nodes. As shown in Fig 6, our model slightly 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. 8. Hence our model bounds the performance expected for the whole system.

Fig. 8 also shows the impact of swarm size on the server capacity. The slow peers stay in the system longer than the fast peers, hence they benefit more from the upload capacity of the server. In addition, we observe 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. For small swarms, increasing the server capacity can improve download times, but only up to a certain point, i.e. 40 MBps for 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

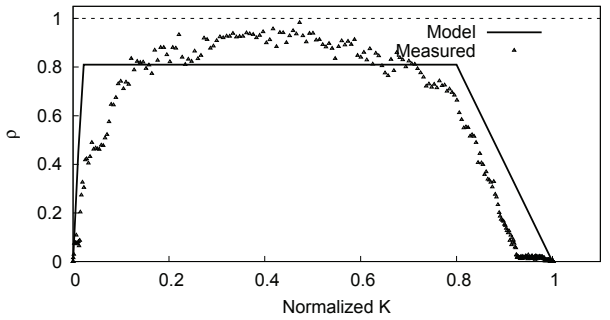


Fig. 9: ρ for 50-node FairTorrent swarm ($C_s = 256$ kBps).

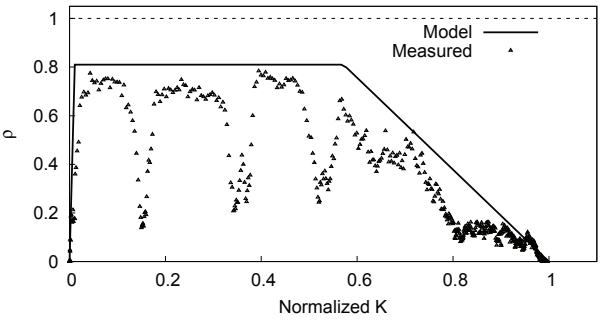


Fig. 10: Starvation in 110-node FairTorrent swarm ($C_s = 256$ kBps).

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 different server and peer upload bandwidths, and file sizes, among others.

6.3 Modeling Other Protocols

In principle, our model described in Section 4.1 can be applied to other p2p protocol by deriving the protocol specific parameters. To illustrate this, we present our model results for BT swarms with some nodes that start seeding after finishing their download (BTSeed), and for FairTorrent (FT) [30], a newly proposed protocol with a different incentive mechanism.

To model BTSeed, we relax the pessimistic assumption of peers leaving after their download and we show that the model captures swarm characteristics accurately. The dynamics of the swarms is similar to BT for the first two download phases of start-up and maximum utilization. The main difference is in the end-game phase because the remaining peers cannot fully utilize the upload bandwidth of the new seeds. Hence, the bottleneck in reaching maximum utilization is not the uploading bandwidth, but the downloading capacity of the remaining peers. For BTSeed, the average swarm download time decreases with the increase of the number peers that remain as seeds. The utilization profile in all our experiments matches the original profile shown in Fig. 3. Lastly, an average error of 8.9% in estimating the average download time is comparable to our original model.

To further demonstrate the flexibility of our model, we model FairTorrent, which has a different incentive scheme. FT effectively unchokes all peers, and peers are serviced in a manner to minimize the deficit between the amount of data uploaded and downloaded with respect to each peer. In our model, we assume a maximum deficit of one block, i.e., that a peer will only upload to another peer when the difference between the number of bytes uploaded to and downloaded from that peer is less than one block. In this way, the protocol ensures the uploads and downloads between each pair of peers are matched. Considering this, the model parameters for FT are:

$$\alpha_{FT} = \frac{\rho_{max} C}{C_s M \ln(\frac{C}{C_s}\rho_{max} + 1)} \tag{12}$$

$$\beta_{FT} = \alpha + \frac{(M-1)}{M}\frac{C_s + \rho_{max} C}{C_s + C} \tag{13}$$

Model derivation for FT is in Appendix D.2.

Fig. 9 shows an experiment with 50 nodes, a server capacity of 256 kBps and a peer capacity of 128 kBps. We observed a trapezoidal shape for ρ similar to that for BT. We ran 70 experiments, varying the number of peers and the server bandwidth. We found that 44 experiments had an average error of 14% in the estimated average download time, while the remaining 26 experiments had an average error of 45%.

Upon further investigation, we found that in the experiments with the large errors, FairTorrent suffered from starvation. An example trace for one of these experiments with 110 nodes is shown in Fig. 10. In this experiment, the server capacity was 256 kBps and the peer capacity was 128 kBps. Starvation periodically causes significant drops in the utilization, which resulted in actual download times that were significantly slower than those predicted by our model. Starvation was caused by the strict condition in the block exchange mechanism, causing some peers to end up in a state where they do not upload to the other peers even though they have sufficient upload bandwidth. This suggested that the enforcement of a strict fairness policy in FT could degrade performance. Probing further, we found that if we removed the upper limit on the maximum deficit, starvation becomes less likely, but it could still happen and was relatively common. This phenomenon was not reported by the authors of FairTorrent [30].

7 Conclusion

Our measurement study of BitTorrent on PlanetLab shows that the utilization of the available bandwidth of peers has a fixed pattern characterized by three phases over the course of a download. Based on this insight from measurement, we proposed an analytical model for studying the impact of bandwidth utilization on the performance of p2p file distribution systems with flash crowd. Leveraging the insight that the utilization of available peer bandwidth for the

system varies as the download progresses, we estimate the download rate and average download time. Validation using PlanetLab for different BT swarms showed that the download rate variation over time closely follows the predicted model profile. For homogeneous and heterogeneous swarms up to three classes of peers, the error for average download time is around 10%. The accuracy of our model can be further improved by considering the server distribution policies, network connectivity and conditions, among others.

Applying our model, we observed a number of insights in how p2p protocols perform during flash crowd. We showed that the download time expected by users in a flash crowd situation does not increase significantly when the swarm size is larger than 50 peers. Secondly, server bandwidth provisioning during flash crowds is a challenging problem because it takes a significant increase in server bandwidth to achieve small improvements in the quality of service. However, for homogeneous swarms, our model shows that by slightly relaxing the requirements on download time by 10%, the required server capacity can potentially be reduced by 40%. This represents a significant saving. In a heterogeneous system, 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. We leave the analysis of the impact of the server capacity on the utilization of available bandwidth as future work.

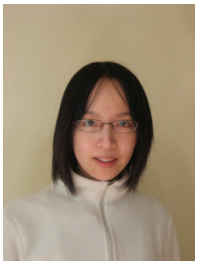
REFERENCES

[1] BitTorrent. http://www.bittorrent.org.
[2] Blizzard Entertainment. World of Warcraft. http://www.blizzard.co.uk/wow/faq/bittorrent.shtml.
[3] N. Carlsson, D. L. Eager, A. Mahanti. Using Torrent Inflation to Efficiently Serve the Long Tail in Peer-Assisted Content Delivery Systems. *Proc. of IFIP Networking*, 2010.
[4] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, M. Bowman. PlanetLab: An Overlay Testbed for Broad-Coverage Services. *Computer Communication Review*, 33:3–12, 2003.
[5] B. Cohen. Incentives Build Robustness in BitTorrent. *Workshop on Economics of Peer-to-Peer Systems*, 2003.
[6] L. D'Acunto, T. Vinko, J. Pouwelse. Do BitTorrent-Like VoD Systems Scale under Flash-crowds? *Proc. of IEEE International Conference on Peer-to-Peer Computing*, 2010.
[7] G. Dán, N. Carlsson. Dynamic Swarm Management for Improved BitTorrent Performance. *Proc. of International Conference on Peer-to-peer Systems*, 2009.
[8] S. Das, S. Tewari, L. Kleinrock. The Case for Servers in a Peer-to-Peer World. *Proc. of IEEE International Conference on Communications*, 2006.
[9] B. Fan, J. C. S. Lui, D.-M. Chiu. The Design Trade-offs of BitTorrent-like File Sharing Protocols. *Transactions on Networking*, 2009.
[10] Fedora. Fedora Project Bittorrent Tracker. http://torrent.fedoraproject.org/.
[11] Z. Ge, D. R. Figueiredo, S. Jaiswal, J. Kurose, D. Towsley. Modeling Peer-peer File Sharing Systems. *Proc. of IEEE Conference on Computer Communications*, 2003.
[12] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, X. Zhang. A Performance Study of BitTorrent-like Peer-to-peer Systems. *IEEE Journal on Selected Areas in Communications*, 25:155–169, 2007.
[13] A. Iosup, P. Garbacki, J. A. Pouwelse, D. Epema. Analyzing BitTorrent: Three Lessons from One Peer-Level View. *Proc. of ASCI Conference*, 2005.
[14] M. Izal, U. G. Keller, E. Biersack, P. Felber, A. Hamra, G. L. Erice. Dissecting BitTorrent: Five Months in a Torrent's Lifetime. *Passive and Active Measurement Workshop*, 2004.
[15] R. Kumar, K. Ross. Peer-Assisted File Distribution: The Minimum Distribution Time. *IEEE Workshop on Hot Topics in Web Systems and Technologies*, 2006.
[16] A. Legout, N. Liogkas, E. Kohler, L. Zhang. Clustering and Sharing Incentives in BitTorrent Systems. *ACM SIGMETRICS Performance Evaluation Review*, 35:301–312, 2007.
[17] F. Lehrieder, G. Dán, T. Hossfeld, S. Oechsner, V. Singeorzan. The Impact of Caching on BitTorrent-Like Peer-to-Peer Systems. *Proc. of IEEE International Conference on Peer-to-Peer Computing*, 2010.
[18] B. Leong, Y. Wang, S. Wen, C. Carbunaru, Y. M. Teo, C. Chang, T. Ho. Improving Peer-to-Peer File Distribution: Winner Doesn't Have to Take All. *Asia-Pacific Workshop on Systems*, 2010.
[19] M. Li, J. Yu, J. Wu. Free-riding on BitTorrent-like Peer-to-peer File Sharing Systems: Modeling Analysis and Improvement. *Transactions on Parallel and Distributed Systems*, 19:954–966, 2008.
[20] W. C. Liao, F. Papadopoulos, K. Psounis. Performance Analysis of BitTorrent-like Systems with Heterogeneous Users. *Performance Evaluation*, 64:876–891, 2007.
[21] D. S. Menasché, A. A. Rocha, E. A. de Souza e Silva, R. M. Leão, D. Towsley, A. Venkataramani. Estimating Self-sustainability in Peer-to-peer Swarming Systems. *Performance Evaluation*, 67:1243–1258, 2010.
[22] M. Meulpolder, J. A. Pouwelse, D. H. J. Epema, H. J. Sips. Modeling and Analysis of Bandwidth-inhomogeneous Swarms in BitTorrent. *Proc. of Peer-to-Peer Computing*, 2009.
[23] V. Misra, P. Barford, M. S. Squillante. Incentivizing Peer-assisted Services: A Fluid Shapley Value Approach. *ACM SIGMETRICS Performance Evaluation Review*, 2010.
[24] PeerSim. http://peersim.sourceforge.net.
[25] R. S. Peterson, E. G. Sirer. Antfarm: Efficient Content Distribution with Managed Swarms. *Proc. of USENIX Symposium on Networked Systems Design and Implementation*, 2009.
[26] M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy, A. Venkataramani. Do Incentives Build Robustness in BitTorrent? *Proc. of USENIX Symposium on Networked Systems Design and Implementation*, 2007.
[27] J. A. Pouwelse, P. Garbacki, D. H. J. Epema, H. J. Sips. The Bittorrent P2p File-Sharing System: Measurements And Analysis. *The International Workshop on Peer-to-Peer Systems*, 2005.
[28] D. Qiu, R. Srikant. Modeling and Performance Analysis of BitTorrent-like Peer-to-peer Networks. *Proc. of ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, 2004.
[29] I. Rimac, A. Elwalid, S. Borst. On Server Dimensioning for Hybrid P2P Content Distribution Networks. *Proc. of IEEE International Conference on Peer-to-Peer Computing*, 2008.
[30] A. Sherman, J. Nieh, C. Stein. FairTorrent: Bringing Fairness to Peer-to-peer Systems. *Proc. of ACM International Conference on Emerging Networking Experiments and Technologies*, 2009.
[31] F. Simatos, P. Robert, F. Guillemin. A Queueing System for Modeling a File Sharing Principle. *ACM SIGMETRICS Performance Evaluation Review*, 36:181–192, 2008.
[32] Y. Sun, F. Liu, B. Li, B. Li. Peer-assisted Online Storage and Distribution: Modeling and Server Strategies. *Proc. of International Workshop on Network and Operating Systems Support for Digital Audio and Video*, 2009.
[33] Y. Tian, D. Wu, K. Ng. Modeling, Analysis and Improvement for BitTorrent-like File Sharing Networks. *Proc. of IEEE Conference on Computer Communications*, 2006.
[34] Ubuntu. Downloads. http://torrent.ubuntu.com:6969/.
[35] X. Yang, 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.
[36] B. Zhang, A. Iosup, J. Pouwelse, D. Epema. Identifying, Analyzing, and Modeling Flashcrowds in BitTorrent. *Proc. of IEEE International Conference on Peer-to-Peer Computing*, 2011.

Cristina Carbunaru is pursuing her Ph.D. in Computer Science at the National University of Singapore. She received her Bachelor's degree in Computer Engineering in 2007 at Politehnica University of Bucharest. Her research interests include analytic modeling and performance analysis for distributed systems.

Yong Meng Teo is an Associate Professor of Computer Science at the National University of Singapore and a Visiting Professor at the Shanghai Advanced Research Institute, Chinese Academy of Sciences in China. He received his B.Tech (1st Class Hons) in computer science from the University of Bradford in UK, and his Master and Ph.D. in computer science from the University of Manchester in UK. His research interests are in parallel and distributed computing, performance analysis and systems modeling and simulation.

Ben Leong is an Assistant Professor of Computer Science at the School of Computing, National University of Singapore. He received his Ph.D., M.Eng. and S.B. degrees from the Massachusetts Institute of Technology in 2006, 1997 and 1997 respectively. His research interests are in the areas of computer networking and distributed systems.



Tracey Ho is an Assistant Professor in Electrical Engineering and Computer Science at the California Institute of Technology. She received a Ph.D. (2004) and B.S. and M.Eng degrees (1999) in Electrical Engineering and Computer Science (EECS) from the Massachusetts Institute of Technology (MIT). She was a co-recipient of the 2009 Communications and Information Theory Society Joint Paper Award. Her primary research interests are in information theory, network coding and communication networks.