Available online at www.sciencedirect.com
ScienceDirect
Performance Evaluation 64 (2007) 876–891
PERFORMANCE EVALUATION
An International Journal
www.elsevier.com/locate/peva

Performance analysis of BitTorrent-like systems with heterogeneous users

Wei-Cherng Liao[a,*], Fragkiskos Papadopoulos[a], Konstantinos Psounis[b]

[a] Department of Electrical Engineering, University of Southern California, Los Angeles, USA
[b] Department of Electrical Engineering and Computer Science, University of Southern California, Los Angeles, USA

Available online 21 June 2007

Abstract

Among all peer-to-peer (P2P) systems, BitTorrent seems to be the most prevalent one. This success has drawn a great deal of research interest on the system. In particular, there have been many lines of research studying its scalability, performance, efficiency, and fairness. However, despite the large body of work, there has been no attempt mathematically to model, in a heterogeneous (and hence realistic) environment, what is perhaps the most important performance metric from an end user's point of view: the average file download delay.

In this paper we propose a mathematical model that accurately predicts the average file download delay in a heterogeneous BitTorrent-like system. Our model is quite general, has been derived with minimal assumptions, and requires minimal system information. Then, we propose a flexible token-based scheme for BitTorrent-like systems that can be used to tradeoff between overall system performance and fairness to high bandwidth users, by properly setting its parameters. We extend our mathematical model to predict the average file download delays in the token- based system, and demonstrate how this model can be used to decide on the scheme's parameters that achieve a target performance/fairness.

Keywords: P2P networks; BitTorrent; Performance analysis; Token-based scheme; Fairness/delay tradeoff

1. Introduction

Peer-to-peer (P2P) systems have provided a powerful infrastructure for large-scale distributed applications, such as file sharing. As a result, they have become very popular. For example, 43% of the Internet traffic is P2P traffic [1]. Among all P2P systems, BitTorrent seems to be the most prevalent one. In particular, more than 50% of all P2P traffic is BitTorrent traffic [2].

The BitTorrent system is designed for efficient large-scale content distribution. The complete BitTorrent protocol can be found in [3]. We summarize the main functionality here. BitTorrent groups users by the file in which they are interested. In each group there exists at least one user, called seed, who has the complete file of interest. The seed is in charge of disseminating the file to other users, called leechers, who do not have the file. When disseminating the

* Corresponding author. Tel.: +1 213 740 3963; fax: +1 213 740 4418.
E-mail addresses: weicherl@usc.edu (W.-C. Liao), fpapadop@usc.edu (F. Papadopoulos), kpsounis@usc.edu (K. Psounis).

0166-5316/$ - see front matter
doi:10.1016/j.peva.2007.06.008

file, BitTorrent partitions the whole file into a large number of blocks and then the seed starts uploading blocks to its neighbors. Meanwhile, users of the group exchange the blocks they have with their neighbors. When a user has all the blocks of the file, he/she finishes the download process and becomes a potential seed.

There are at least three features that make BitTorrent successful. First, BitTorrent breaks a complete file into blocks and disseminates the file by sending blocks instead of sending the complete file. In this way, users, who have a partial file, can exchange their blocks with their neighbors without the help of the seed. As a result, the service capacity of the system is enlarged because every participating user can contribute to the system even if he/she only has a partial file. Second, BitTorrent uses the local rarest first (LRF) block selection algorithm to disseminate blocks, which means users will prefer to download the *rarest* block among their neighbors. After a user downloads the "rarest" block, he/she can disseminate this block to other users and thus can increase the availability of this block. It has been shown that the LRF algorithm can efficiently enlarge the service capacity and prevent the last block problem [4]. The last important feature of BitTorrent is its rate-based Tit-for-Tat (TFT) unchoking scheme. In the rate-based TFT unchoking scheme, a user will provide uploads to four neighbors who provide him/her with the highest download rates and to one more, a randomly selected neighbor, via a process called optimistic unchoking. This scheme successfully discourages freeriders in the BitTorrent system because freeriders will keep getting choked if they do not provide uploads to other users. Because of all the above features, BitTorrent provides a fast and efficient infrastructure for large-scale content distribution.

Because of the prevalence and the success of BitTorrent, there is a large body of work studying various aspects of the BitTorrent system, such as its performance analysis [5–12], incentive schemes for it [13,14], traffic measurements [2,15–17], and fairness issues [4,18]. However, despite this large body of research, there has been no attempt mathematically to model, in a heterogeneous and hence realistic environment, what is perhaps the most important performance metric from an end user's point of view: the average file download delay.

Our main contribution in this paper is a simple mathematical model that accurately predicts the average file download delay in a heterogeneous BitTorrent-like system, where users may have different upload/download capacities. Despite its simplicity, our model is quite general, it has been derived with minimal assumptions, and requires minimal system information.

Our second contribution is that we propose a token-based TFT scheme, which is very simple and flexible. In the proposed scheme, which is inspired by our prior work on incentive schemes for P2P systems [19,20], users use tokens as a means to trade blocks. Each user maintains a token table which keeps track of the amount of tokens his/her neighbors possess. A user increases his/her neighbor's tokens by K_{up} for every byte he/she downloads from the neighbor. On the other hand, the user decreases a neighbor's tokens by K_{down} for every byte he/she uploads to the neighbor under study. A user would upload a block to his/her neighbor only if the neighbor has sufficient tokens to perform the download.

We show that the proposed scheme can be used to tradeoff between high overall system performance and fairness to high bandwidth users, by properly setting its parameters K_{down} and K_{up}. In particular, we show that under the appropriate parameter tuning, high bandwidth users will provide more uploads than usual to low bandwidth users which tends to reduce the overall download delay. This, however, comes at the expense of making high bandwidth users download at a slower rate than they usually do. We extend our mathematical model to predict the average file download delays in this system, and demonstrate how the model can be used to decide on the values of K_{down} and K_{up} that achieve a target system performance/fairness.

The rest of this paper is organized as follows: in Section 2 we briefly discuss related work. In Section 3 we review the current BitTorrent implementation in more detail, and provide a detailed description of the proposed token-based scheme. In Section 4 we present the mathematical model that predicts the performance of BitTorrent-like systems, and then we extend the model to predict performance when the token-based scheme is used. In Section 5 we present extensive simulation results in order to validate the accuracy of our model. In the same section we also demonstrate how the model can be used to decide on the scheme parameters that achieve a target tradeoff between overall system performance and fairness to high bandwidth users. Conclusions and future work directions follow in Section 6.

2. Related work

B. Cohen, the author of BitTorrent, gives a thorough introduction to the BitTorrent system in [21]. The paper describes the BitTorrent protocol, the system architecture and the incentive scheme built in the BitTorrent system. In addition, there is a large body of work reporting the efficiency and the popularity of BitTorrent [12,13,16]. Although there are some studies, *e.g.* [13,14], indicating that skillful freeriders can still benefit from the system against the built-in incentive scheme, BitTorrent in general has successfully motivated users to share their resources.

To the best of our knowledge, [5] is the first published work to provide a mathematical model for the BitTorrent system. The paper proposes a fluid model to describe how the population of seeds and leechers evolves in the BitTorrent system. [7,8,10] extend the above model to study BitTorrent's performance under different user behaviors and different arrival processes. Further, [6,9] extend this model to study BitTorrent's performance under heterogeneous environments. In [11], the authors propose a model to study the peer distribution in BitTorrent and they use a dying process to study the file availability. Also, [22] provides a queueing model for P2P file-sharing systems, [23] uses a branching process to study the capacity of P2P systems, and [24] proposes a fluid model to study a generic P2P file-sharing system.

Despite the large body of work on modelling BitTorrent's performance, the majority of studies, *e.g.* [5,7,8,10], consider homogeneous network environments only, where users have the same capacities. This is clearly an unrealistic assumption given the Internet's heterogeneity. Although the studies in [6,9] consider network heterogeneity, their analysis ignores an important feature of the BitTorrent system, its rate-based TFT scheme, which as we mentioned earlier, is one of the main features responsible for the system's great success. Further, these studies do not provide any simulation or experimental results to verify the validity of their model. In this paper we consider a heterogeneous BitTorrent-like system, where users can be grouped into two categories: (i) high bandwidth (**H-BW**) users, who have high upload link capacities, and (ii) low bandwidth (**L-BW**) users, who have low upload link capacities. Then, we propose a mathematical model that can predict the average download delay of each class of users and for the whole system. Our model is quite simple and general. Further, we verify via extensive simulations that it is remarkably accurate.

The work in [4] proposes a block-based TFT scheme. Our proposed token-based TFT scheme, which is inspired by our prior work on incentive schemes for P2P systems [19,20], degenerates to the scheme in [4] when $K_{\text{up}} = K_{\text{down}}$. Hence, our scheme is much more general and flexible. Further, the work in [4] studies the performance of the block-based TFT scheme only via simulations. Here, we extend our mathematical model to predict the performance of the token-based scheme for a general $\frac{K_{\text{up}}}{K_{\text{down}}}$ ratio. Finally, we show how our model can be used to decide on the scheme parameters that achieve a target tradeoff between overall system performance and fairness to H-BW users.

3. The BitTorrent system and the proposed token-based scheme

3.1. The BitTorrent system

We now describe in detail the main functionality of the BitTorrent system. Recall that BitTorrent groups users by the file in which they are interested. When a user is interested in joining a group, he/she first contacts the tracker, a specific host that keeps track of all the users currently participating in the group. The tracker responds to the user with a list containing the contact information of L randomly selected peers. (Typical values for L are 40–60 [3].) After receiving the list, the user establishes a TCP connection to each of these L peers, to which we refer as the user's *neighbors*.

As mentioned earlier, when disseminating the file, BitTorrent partitions the whole file into a number of blocks. Neighbors exchange block availability information and messages indicating interest in blocks. The BitTorrent protocol uses a rate-based TFT scheme to determine to which neighbors a user should upload blocks. The rate-based TFT scheme proceeds as follows: time is slotted into 10 s intervals and each such time-interval is called an *unchoking period*. At the end of each unchoking period a user makes a *choking/unchoking* decision. The choking/unchoking decision proceeds as follows: first, the user computes for each of the neighbors that are interested in downloading a block from him/her, the average download rate that he/she receives during the last 20 s. Then, he/she selects to provide uploads to, i.e. to *unchoke*, the four neighbors who provided him/her with the best download rates, with ties broken arbitrarily. (Similarly, if the user chooses not to provide uploads to a neighbor, we say that the neighbor is choked.) Finally, he/she also randomly selects another neighbor to whom to provide uploads. This last (random) selection process is called *optimistic unchoking*. Hence, at any time instance a user is concurrently uploading to five neighbors. The following rules are also adopted by the scheme.

Let us call the neighbor that was selected at the last optimistic unchoking, an *optimistic unchoking neighbor*, and suppose that the last optimistic unchoking (and hence the end of the last unchoking period) took place at time t_1 seconds. Now, suppose that the end of another unchoking period occurs at some time t_2 seconds. (Clearly, $t_2 \geq t_1 + 10$ s.) Then, if at time t_2 the optimistic unchoking neighbor belongs to the set of the four neighbors who provide to the user the best download rates (and hence they will be unchoked), the user performs a new optimistic unchoking. Otherwise: (i) if $t_2 < t_1 + 30$ s, the user does not choke the optimistic unchoking neighbor and does not perform a new optimistic unchoking, and (ii) if $t_2 \geq t_1 + 30$ s, the user chokes the optimistic unchoking neighbor and performs a new optimistic unchoking. We call this 30 s time-interval an *optimistic unchoking period*.

This TFT scheme successfully discourages free-riders because they will keep getting choked if they do not provide uploads to their neighbors. Further, it gives the opportunity to new users to start downloading from the system even if they do not have enough blocks to exchange, in which case the download rate they provide is low. Finally, notice that the scheme allows a user to discover good neighbors, i.e. neighbors who provide him/her with high download rates, and to exchange data with them. Therefore, users who have high upload link capacities tend to exchange data with a larger number of high-capacity users; and users with low upload link capacities tend to exchange data with a larger number of low-capacity users. Hence, in a sense the system is designed to be fair to each class of user.

3.2. The proposed token-based scheme

The process by which a new user discovers neighbors in the proposed token-based scheme is exactly the same as the original BitTorrent system. Further, again, the file is partitioned into blocks and neighbors exchange block availability information and messages indicating an interest in blocks.

As mentioned earlier, in the token-based system users use tokens as a means to trade blocks. In particular, each user maintains a token table, which keeps track of the amount of tokens his/her neighbors possess. When the user uploads X_{up} bytes to a neighbor, he/she decreases the neighbor's tokens by $K_{\text{down}} X_{\text{up}}$. On the other hand, the user increases a neighbor's tokens by $K_{\text{up}} X_{\text{down}}$ if he/she downloads X_{down} bytes from the neighbor under study. Notice that a user does not have access to his/her amount of tokens since this is maintained by his/her neighbors.

Under the proposed scheme each user decides to which (of the interested) neighbors he/she will upload blocks, every 10 s. This is equal to the unchoking period in the original BitTorrent system. In particular, every 10 s the user first checks which of his/her neighbors have enough tokens to perform the download of a block. If there are more than five neighbors having enough tokens, then the user randomly selects five of them to upload to, which is equal to the number of peers a user provides uploads to in the original BitTorrent system. If five or fewer neighbors have enough tokens the user provides uploads only to them. If a neighbor runs out of tokens while downloading from the user, then the user stops uploading to the neighbor immediately after the current block transfer is complete, and randomly selects to upload to some other neighbor who has enough tokens. Finally, we initialize the token table of each user with an amount of tokens that suffices to download one block. The reason for giving initial tokens is to allow users download data when they first join the system.

Note that K_{up} and K_{down} are relative values. Therefore, the proposed scheme actually has only one design parameter. We will show that for $K_{\text{up}} = K_{\text{down}}$ the proposed token-based scheme has approximately the same performance, and it is as fair as the original BitTorrent system. Finally, we will also show that as K_{up} increases the overall system performance of the token-based scheme can become significantly better than that of the original BitTorrent system by sacrificing some fairness towards high-capacity users. In particular, high-capacity users will end up providing uploads to the system at a faster rate than the download rate they receive.

4. A mathematical model for the performance of BitTorrent-like systems

In this section we propose a mathematical model to study the performance of BitTorrent-like systems in steady state, where the number of peers in the system does not change. In particular, we focus on studying the average file download delay, which is the time difference between the moment that a user joins a group and the moment that the user downloads the complete file.

As mentioned earlier, in real P2P systems users have heterogeneous capacities. We incorporate this fact in our analysis in order to make it more realistic and general. In particular, we assume that there exist two classes of user: (i) high-bandwidth (**H-BW**) users, who have a high-upload link capacity, and (ii) low-bandwidth (**L-BW**) users, who have a low-upload link capacity.[1] We denote by α the percentage of L-BW users in the system. Further, we assume that the number of leechers is much larger than the number of seeds, as this is usually the case in real systems [16,17]. We start our analysis with the original BitTorrent system and then proceed with the proposed token-based system.

4.1. A mathematical model for the original BitTorrent system

4.1.1. Computing the download rates of H-BW and L-BW users

Consider a H-BW user and denote by n^d_{HH} and n^d_{HL} the steady state average number of H-BW and L-BW neighbors respectively from which this user is downloading, and by D_{HH} and D_{HL} the corresponding average download rates. Similarly, consider a L-BW user and denote by n^d_{LH} and n^d_{LL} the steady state average number of H-BW and L-BW neighbors respectively from which this user is downloading, and by D_{LH} and D_{LL} the corresponding average download rates. Now, let $R_{\mathrm{down}H}$ and $R_{\mathrm{down}L}$ be the aggregate download rate of a H-BW and a L-BW user respectively. It is easy to see that:

$$R_{\mathrm{down}H} = n^d_{\mathrm{HH}} D_{\mathrm{HH}} + n^d_{\mathrm{HL}} D_{\mathrm{HL}}, \tag{1}$$

$$R_{\mathrm{down}L} = n^d_{\mathrm{LH}} D_{\mathrm{LH}} + n^d_{\mathrm{LL}} D_{\mathrm{LL}}. \tag{2}$$

Now, denote by n^u_{HH} and n^u_{HL} the steady state average number of H-BW and L-BW neighbors respectively to which a H-BW user is uploading, and let U_{HH} and U_{HL} be the corresponding average upload rates. Similarly, denote by n^u_{LH} and n^u_{LL} the steady state average number of H-BW and L-BW neighbors respectively to which a L-BW user is uploading, and by U_{LH} and U_{LL} the corresponding average upload rates. Further, let $R_{\mathrm{up}H}$ and $R_{\mathrm{up}L}$ be the aggregate upload rate of a H-BW and a L-BW user respectively. As before, we can write:

$$R_{\mathrm{up}H} = n^u_{\mathrm{HH}} U_{\mathrm{HH}} + n^u_{\mathrm{HL}} U_{\mathrm{HL}}, \tag{3}$$

$$R_{\mathrm{up}L} = n^u_{\mathrm{LH}} U_{\mathrm{LH}} + n^u_{\mathrm{LL}} U_{\mathrm{LL}}. \tag{4}$$

In order to be able to predict the download delays, we first need to compute $R_{\mathrm{down}H}$ and $R_{\mathrm{down}L}$. Hence, we need to calculate the values of the parameters n^d_{HH}, n^d_{HL}, n^d_{LH}, n^d_{LL}, D_{HH}, D_{HL}, D_{LH}, and D_{LL}. To do so, we will first compute the values of n^u_{HH}, n^u_{HL}, n^u_{LH}, n^u_{LL}, U_{HH}, U_{HL}, U_{LH}, and U_{LL} and then relate them to the aforementioned parameters.[2]

In order to compute n^u_{HH}, n^u_{HL}, n^u_{LH}, n^u_{LL}, U_{HH}, U_{HL}, U_{LH}, and U_{LL}, we first need to find, in addition to Eqs. (3) and (4), six more relations. In this way we will have a system comprising of eight equations and eight unknowns.[3]

First, recall that at any time instance, a user in BitTorrent is uploading to five of its neighbors. Hence, we have:

$$n^u_{\mathrm{HH}} + n^u_{\mathrm{HL}} = 5, \tag{5}$$

$$n^u_{\mathrm{LH}} + n^u_{\mathrm{LL}} = 5. \tag{6}$$

Let $C_{\mathrm{up}H}/C_{\mathrm{down}H}$ and $C_{\mathrm{up}L}/C_{\mathrm{down}L}$ be the upload/download link capacity of H-BW and L-BW users respectively. Further, assume that a user's download link capacity is larger than or equal to his/her upload link capacity. Therefore, the system's bottlenecks are the upload links and we can assume that these are fully utilized.[4] This means that $R_{\mathrm{up}H} = C_{\mathrm{up}H}$ and that $R_{\mathrm{up}L} = C_{\mathrm{up}L}$.

Since peer-to-peer traffic is transferred via TCP connections, we assume that the upload capacity of a user will be fairly shared among concurrent upload connections, *if* the maximum possible download rate of *each* connection is larger or equal to the fair share. For L-BW users this is always the case since $C_{\mathrm{down}H} > C_{\mathrm{up}L}$, and $C_{\mathrm{down}L} \geq C_{\mathrm{up}L}$, and we can state the following lemma whose proof is straightforward:

[1] The studies in [4,25] divide the users of a real P2P system according to their upload link capacities into four classes. Here, we assume two classes of user only, for ease of exposition. Our analysis can be extended along the same lines for more classes of user.

[2] Computing these parameters first is easier. This is because, it is the rules according to which a user chooses a neighbor to provide *uploads to*, that are explicitly defined in the BitTorrent protocol.

[3] In general, if there are n classes of users, one would need to solve a system of $(n + n) \cdot n = 2n^2$ equations. This is because each class $C \in \{1 \cdots n\}$ is characterized by n variables dictating the number of users from each class to which a member of class C is uploading on average, and n corresponding upload rates.

[4] This is not an unrealistic assumption. Common Internet access technologies, such as dial-up, DSL, cable modem, and ethernet, satisfy this assumption [25]. Further, this assumption has also been made in many studies on peer-to-peer networks, *e.g.* see [10] and references therein, and it is in accordance with measurement studies of BitTorrent systems, *e.g.* see [12,16].

Lemma 1.

$$U_{\mathrm{LL}} = U_{\mathrm{LH}} = \frac{C_{\mathrm{up}L}}{5}. \tag{7}$$

We now turn our attention to the upload rates that a H-BW user provides. At any time instance, a L-BW user is downloading on average from n^d_{LL} L-BW neighbors. We define the *spare* download capacity of this user as $C_{\mathrm{down}L} - n^d_{\mathrm{LL}} D_{\mathrm{LL}}$.[5] Therefore, the upload rate that a H-BW user can provide to a L-BW user is given by the following lemma:

Lemma 2.

$$U_{\mathrm{HL}} = \min\left(\frac{C_{\mathrm{up}H}}{5}, C_{\mathrm{down}L} - n^d_{\mathrm{LL}} D_{\mathrm{LL}}\right) = \min\left(\frac{C_{\mathrm{up}H}}{5}, C_{\mathrm{down}L} - n^u_{\mathrm{LL}} U_{\mathrm{LL}}\right). \tag{8}$$

Proof. If the spare capacity of the L-BW user is larger than his/her fair share ($\frac{C_{\mathrm{up}H}}{5}$), the user will be downloading from the H-BW user at an average rate equal to his/her fair share. Otherwise, the user will be downloading at an average rate equal to his/her spare capacity. Further, since the total download rate from L-BW users to L-BW users equals the total upload rate from L-BW users to L-BW users, $n^d_{\mathrm{LL}} D_{\mathrm{LL}} = n^u_{\mathrm{LL}} U_{\mathrm{LL}}$. □

Now, note that once we know the values for n^u_{HH} and n^u_{HL}, the value of U_{HH} will result from Eq. (3). Further, let L be the total number of a user's neighbors and assume that all of these neighbors are interested in a block that the user under study possesses.[6] Finally, denote by Binomial(N, p, k) the probability mass function of a Binomial random variable with parameters N and p, that is, Binomial$(N, p, k) \equiv \binom{N}{k} p^k (1-p)^{(N-k)}$. Then, n^u_{HL} (the average number of L-BW users that a H-BW user provides uploads to) is given by the following lemma:

Lemma 3.

$$n^u_{\mathrm{HL}} = \sum_{k=0}^{L} n(k)\mathrm{Prob}\{\textit{have } k \text{ H-BW } \textit{neighbors out of } L\}, \tag{9}$$

where:

$$n(k) = \begin{cases} \dfrac{L-k}{L-4} & \textit{if } k \geq 5, \\ 5-k & \textit{otherwise} \end{cases}$$

and:

$\mathrm{Prob}\{\textit{have } k \text{ H-BW } \textit{neighbors out of } L\} = \mathrm{Binomial}(L, 1-\alpha, k)$.

Proof. First, recall that α is the percentage of L-BW users in the system. Since the neighbors' list consists of a random selection of H-BW and L-BW users, it is easy to see that Prob{have k H-BW neighbors out of L} $=$ Binomial$(L, 1-\alpha, k)$.

Now, let us consider a H-BW user, say user j, and let $k \leq L$ be the number of j's H-BW neighbors. Since j provides uploads to five of his/her neighbors, we distinguish two cases: (i) $k \geq 5$, and (ii) $k < 5$. First, consider case (i) and recall how BitTorrent's TFT scheme works (see Section 3). It is easy to see that in this case, j may be uploading to at most one L-BW user at any time instance. This L-BW user is randomly selected (via optimistic unchoking) with

5 Note that in general, because of BitTorrent's TFT strategy (see Section 3), a L-BW user that has been selected from a H-BW user via optimistic unchoking, will be downloading from the H-BW user for a time duration equal to the optimistic unchoking period. When the optimistic unchoking period elapses the H-BW user will choke this L-BW user because he/she provides him/her with a low download rate. Therefore, we will be assuming that the probability that the *same* L-BW user is concurrently downloading from two or more H-BW users is quite small. This is not an unrealistic assumption if the number of users in the system is large. (Recall from Section 3 that typical values for L are 40–60.)

6 It has been demonstrated that file sharing in BitTorrent is very effective, i.e., there is a high likelihood that a node holds a block that is useful to its peers, *e.g.* see [4]. This is partially due to the local rarest first (LRF) block selection algorithm that BitTorrent uses to disseminate blocks.

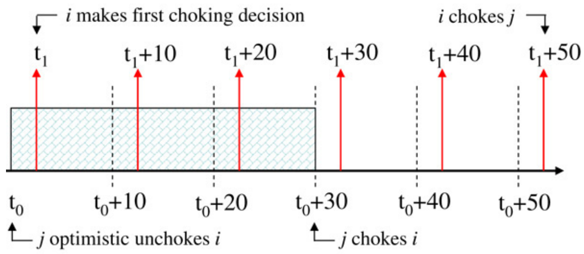


Fig. 1. Time line of optimistic unchoking and choking decision making.

probability $\frac{L-k}{L-4}$. Now, consider case (ii). In this case, j is uploading to exactly $5-k$ L-BW users at any time instance, as he/she does not have any other H-BW neighbor to whom he/she could provide uploads. It is now easy to see that n^u_{HL} is given by Eq. (9). □

Now, recall from Section 3 that the optimistic unchoking period is 30 s, the rate observation window is 20 s, and users make their choking decision every 10 s. Suppose that H-BW user j selects L-BW user i via optimistic unchoking at time t_0, as shown in Fig. 1.

According to BitTorrent's TFT scheme, at time t_0+30 user j will choke i, because i did not provide him/her with a high download rate. Also, suppose that L-BW user i makes his/her first choking decision at time t_1. Clearly, user i will not choke user j at t_1, t_1+10, and t_1+20 because j provides him/her with a higher download rate compared to U_{LL} (the rate by which i is downloading from a L-BW neighbor).[7] Further, user i will choke j at time t_1+50 because the rate observation window is 20 s and user j did not provide anything to i during the period $(t_1+30, t_1+50]$. How about t_1+30 and t_1+40? At t_1+30, the average download rate that i observes from j is $\frac{U_{\text{HL}}(20+t_0-t_1)}{20}$. If this rate is larger than U_{LL}, i will not choke j. Similarly, at t_1+40, the average download rate that i observes from j is $\frac{U_{\text{HL}}(10+t_0-t_1)}{20}$. If this rate is larger than U_{LL}, i will not choke j. Therefore, if N_{unchoke} denotes the number of times that i did not choke j, we can write:

$$N_{\text{unchoke}} = \begin{cases} 3 & \text{if } U_{\text{HL}}\dfrac{(20+t_0-t_1)}{20} < U_{\text{LL}}, \\ 5 & \text{if } U_{\text{HL}}\dfrac{(10+t_0-t_1)}{20} \geq U_{\text{LL}}, \\ 4 & \text{otherwise.} \end{cases}$$

Because users are not synchronized, it makes sense to assume that t_1 is uniformly distributed between t_0 and t_0+10. Hence, we can compute the average number of times $\overline{N}_{\text{unchoke}}$ that i did not choke j. This corresponds to a duration of $10\overline{N}_{\text{unchoke}}$ s.

Now, recall that a H-BW user is uploading to n^u_{HL} L-BW users on average. Therefore, considering the above scenario only, it is easy to see that at any time instance a H-BW user on average downloads from $n^u_{\text{HL}}\frac{10\overline{N}_{\text{unchoke}}}{30}$ L-BW users. Hence, the average number of H-BW users to whom a L-BW user provides uploads (due to the above scenario only) is $\left(\frac{1-\alpha}{\alpha}\right)n^u_{\text{HL}}\frac{10\overline{N}_{\text{unchoke}}}{30}$. We refer to this scenario, as the *optimistic unchoking reward scenario*.

Now, n^u_{LH} (the average number of H-BW users to whom a L-BW user provides uploads) is given by the following lemma:

Lemma 4.

$$n^u_{\text{LH}} = \sum_{i=0}^{L} n(w)\text{Prob}\{\textit{have } w \text{ L-BW } \textit{neighbors out of } L\} + \left(\frac{1-\alpha}{\alpha}\right)n^u_{\text{HL}}\frac{\overline{N}_{unchoke}}{3}, \tag{10}$$

[7] Recall that the probability that two or more H-BW users uploading to the same L-BW user at the same time instance is small. Therefore, i will be always downloading from at least one L-BW neighbor.

where:

$$n(w) = \begin{cases} \dfrac{L-w}{L-4} & \textit{if } w \geq 5, \\ 5-w & \textit{otherwise} \end{cases}$$

and:

$$\mathrm{Prob}\{\textit{have } w \text{ L-BW } \textit{neighbors out of } L\} = \mathrm{Binomial}(L, \alpha, w).$$

Proof. As before, since α is the percentage of L-BW users in the system and the neighbors' list consists of a random selection of H-BW and L-BW users, the probability of having w L-BW neighbors out of L is $\mathrm{Binomial}(L, \alpha, w)$. Further, the second term on the right-hand side of Eq. (10) corresponds to the optimistic unchoking reward scenario. The first term accounts for the number of H-BW users to whom a L-BW user has chosen to upload, just as in the proof of Lemma 3.

In particular, consider L-BW user i, and let $w \leq L$ be the number of i's L-BW neighbors. As before, we distinguish two cases: (i) $w \geq 5$, and (ii) $w < 5$. In case (i) i may be uploading to at most one H-BW user at any time instance. This H-BW user has been selected via optimistic unchoking, with probability $\frac{L-w}{L-4}$, and will be choked after the optimistic unchoking period elapses. This is because the H-BW user, who prefers other H-BW users to whom to upload, will not be uploading to this L-BW user. In case (ii), i has selected to upload to exactly $5 - w$ H-BW users, as he/she does not have any other L-BW neighbor to whom to provide uploads. □

Notice that in Lemma 3, we have not considered the optimistic unchoking reward scenario. This is because if a L-BW user selects via optimistic unchoking a H-BW user to whom to provide uploads, say at time t_0 (see Fig. 1), the H-BW user will choke this L-BW user on his/her first choking decision at time t_1 (see Fig. 1), because the L-BW user does not provide him/her with a high download rate. Therefore, H-BW users do not provide uploads to L-BW users in this case (i.e. L-BW users are not receiving any reward for optimistic unchoking H-BW users.)

Given Eqs. (3)–(10) (and $R_{\mathrm{up}H} = C_{\mathrm{up}H}$, $R_{\mathrm{up}L} = C_{\mathrm{up}L}$), we can now compute n^u_{HH}, n^u_{HL}, n^u_{LH}, n^u_{LL}, U_{HH}, U_{HL}, U_{LH}, and U_{LL}. We now proceed to relate these parameters to n^d_{HH}, n^d_{HL}, n^d_{LH}, n^d_{LL}, D_{HH}, D_{HL}, D_{LH}, and D_{LL}. First, clearly $D_{\mathrm{HH}} = U_{\mathrm{HH}}$, $D_{\mathrm{HL}} = U_{\mathrm{LH}}$, $D_{\mathrm{LH}} = U_{\mathrm{HL}}$, and $D_{\mathrm{LL}} = U_{\mathrm{LL}}$. Further, notice that in any system the total number of upload connections equals the total number of download connections. For example, the total number of upload connections provided by H-BW users to L-BW users equals the total number of download connections that L-BW users receive from H-BW users. Therefore, we can write $n^d_{\mathrm{LH}}\alpha = n^u_{\mathrm{HL}}(1-\alpha)$. Similarly, we can easily relate n^d_{HH}, n^d_{HL}, n^d_{LL} to n^u_{HH}, n^u_{LH}, n^u_{LL}, as follows:

$$n^d_{\mathrm{HH}} = n^u_{\mathrm{HH}},$$
$$n^d_{\mathrm{HL}}(1-\alpha) = n^u_{\mathrm{LH}}\alpha,$$
$$n^d_{\mathrm{LL}} = n^u_{\mathrm{LL}}.$$

We can now compute the average download rate of a H-BW user and a L-BW user using Eqs. (1) and (2), and of course the average download rate across all users.

4.1.2. Estimating the average download delay of H-BW and L-BW users

Fig. 2 shows how the total number of peers in a system with H-BW and L-BW users evolves as a function of time. During the time period $(t_0, t_1]$, users join the system. From t_1 to t_2, both H-BW users and L-BW users are present in the system. Since H-BW users have higher capacities, they depart earlier, by time t_3. Afterwards, only L-BW users are present in the system. Our model computes the download rates for each group of users during the time interval $(t_1, t_2]$. Further, the download rate of L-BW users during the interval $(t_3, t_4]$ is just equal to their upload link capacity since this is fully utilized, as explained earlier. Notice that in this paper, we do not present a model for the transient periods $(t_0, t_1]$ and $(t_2, t_3]$. We make the assumption that $t_0 \approx t_1$ and $t_2 \approx t_3$.[8]

[8] Note that the assumption that $t_1 \approx t_0$ can be justified in a flash crowd scenario where all users join the system in a small time interval. Further, as we shall see, the approximation $t_2 \approx t_3$ does not significantly affect the accuracy of our model.

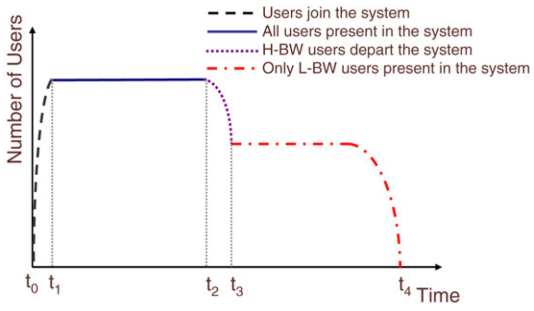


Fig. 2. Evolution of the number of peers: (i) during $(t_0, t_1]$ new users join the system, (ii) during $(t_1, t_2]$ all users are present in the system, (iii) during $(t_2, t_3]$ H-BW users depart the system, and (iv) during $(t_3, t_4]$ only L-BW users are present in the system.

Now, let S be the file size and let T_H and T_L be the average file download delay of a H-BW and a L-BW user respectively. It is easy to see that:

$$T_H = \frac{S}{R_{\text{down}H}}. \tag{11}$$

Further, let S_d be the amount of data that a L-BW user has downloaded when all H-BW users were present in the system. It is easy to see that $S_d = T_H R_{\text{down}L}$. After H-BW users leave the system, the average download rate of L-BW users is just equal to their upload capacity. Hence, the average file download delay of a L-BW user can be expressed as follows:

$$T_L = T_H + \frac{S - S_d}{C_{\text{up}L}}. \tag{12}$$

Note that going from rates to delays is a relatively easy task; it is the computation of rates that is quite involved; and, as already mentioned, our model is the first to compute the rates and, subsequently, the delays for BitTorrent-like systems in the context of heterogeneous users.

4.2. A mathematical model for the token-based system

The model for the token-based system is similar to the model for the original BitTorrent system. In particular, it is easy to see that Eqs. (1)–(6) hold also for the token-based system. Now, let us justify why Eq. (7) holds in this system.

As before, we assume again that the download capacity of a user is larger than or equal to his/her upload capacity. Now, recall that a user earns K_{up} tokens for each byte he/she uploads and spends K_{down} tokens for each byte he/she downloads. For a L-BW user, his/her L-BW neighbors may earn tokens by uploading to him/her at a rate $K_{\text{up}}U_{\text{LL}}$, and they spend tokens by downloading from him/her at a rate $K_{\text{down}}D_{\text{LL}}$. Clearly, to make the token-based system operate properly, we need to have $K_{\text{up}} \geq K_{\text{down}}$. Hence, $K_{\text{up}}U_{\text{LL}} \geq K_{\text{down}}D_{\text{LL}}$ (since $D_{\text{LL}} = U_{\text{LL}}$). Now, consider a H-BW user. The rate that a H-BW user gains tokens by providing uploads to a L-BW user ($K_{\text{up}}U_{\text{HL}}$) is larger than the rate that the user spends tokens by downloading from the L-BW user ($K_{\text{down}}D_{\text{HL}}$), since $K_{\text{up}}U_{\text{HL}} \geq K_{\text{down}}U_{\text{HL}} > K_{\text{down}}U_{\text{LH}} = K_{\text{down}}D_{\text{HL}}$. Therefore, all users will always have enough tokens to download from a L-BW user. Hence, the upload capacity of a L-BW user is fully utilized and Eq. (7) holds true in this system as well. Now, let us see what relations change compared to the original BitTorrent system.

Recall that according to the token-based scheme a user randomly selects to upload to those neighbors who have enough tokens to perform the download. Now, consider a L-BW user. Because all users always have enough tokens to download from a L-BW user, the L-BW user will equally select every peer to whom to provide uploads. Since the total number of upload connections is five, the percentage of H-BW users in the system is $1 - \alpha$, and the neighbor's list consists of a random selection of H-BW and L-BW users, in this system $n^u_{\text{LH}} = 5(1 - \alpha)$.

Before proceeding, consider the scenario where a H-BW user exchanges data with another H-BW user. In this case, the H-BW user's token earning rate ($K_{\text{up}}U_{\text{HH}}$) is greater or equal to the user's token spending rate ($K_{\text{down}}D_{\text{HH}}$) (since $K_{\text{up}} \geq K_{\text{down}}$ and $D_{\text{HH}} = U_{\text{HH}}$). Now, to find the rate by which a H-BW user provides uploads to a L-BW user, we proceed as follows.

First, we assume that $K_{\text{up}} U_{\text{LH}} \geq K_{\text{down}} U_{\text{HL}}$. Under this condition, a L-BW user earns tokens by uploading to a H-BW user at a faster rate than the rate that he/she spends tokens by downloading from the H-BW user. This means that a L-BW user always has enough tokens to download from a H-BW user. Since a H-BW user always has enough tokens to download from a H-BW user as well (as $K_{\text{up}} U_{\text{HH}} \geq K_{\text{down}} D_{\text{HH}}$), the H-BW user cannot distinguish H-BW neighbors from L-BW neighbors, and thus he/she provides uploads to all of his/her neighbors with the same probability. Hence, $n^u_{\text{HL}} = 5\alpha$. Further, U_{HL} in this scenario is given in the following lemma:

Lemma 5.

$$U_{\text{HL}} = \sum_{i=0}^{L} \sum_{k=0}^{i} \min\left(\frac{C_{\text{up}H}}{5}, R_{\text{HL}}(k)\right) P_1(k|i) P_2(i), \tag{13}$$

where:

$$R_{\text{HL}}(k) = \begin{cases} \dfrac{C_{\text{down}L} - n^d_{\text{LL}} U_{\text{LL}}}{k} & \text{if } k > 0, \\ 0 & \text{otherwise} \end{cases}$$

and:

$$P_1(k|i) = \text{Prob}\{\textit{download from k out of i } \text{H-BW } \textit{neighbors}\} = \text{Binomial}\left(i, \frac{5}{L}, k\right),$$
$$P_2(i) = \text{Prob}\{\textit{have i } \text{H-BW } \textit{neighbors out of } L\} = \text{Binomial}(L, 1-\alpha, i).$$

Proof. First, as we have said, a L-BW user always has enough tokens to download from a H-BW neighbor. Hence, his/her download rate is not constrained by the amount of tokens he/she possesses. If a L-BW user is downloading from $k > 0$ H-BW users, the average download rate from each H-BW user is equal to $R_{\text{HL}}(k) = \frac{C_{\text{down}L} - n^d_{\text{LL}} U_{\text{LL}}}{k}$, where $C_{\text{down}L} - n^d_{\text{LL}} U_{\text{LL}}$ is the spare capacity of the L-BW user. However, this rate cannot exceed the maximum average rate that a L-BW user can download from a H-BW user, which is $\frac{C_{\text{up}H}}{5}$. Further, the probability that the L-BW user is downloading from a H-BW neighbor is $\frac{5}{L}$ because each user randomly selects five out of L neighbors to provide uploads to (as every neighbor always has enough tokens). Therefore, given that a L-BW user has i H-BW neighbors, the probability that he/she is downloading from $k \leq i$ of them is Binomial$(i, \frac{5}{L}, k)$. Finally, the probability that the L-BW user has i H-BW neighbors is Binomial$(L, 1-\alpha, i)$. □

Notice that under the aforementioned condition the upload link capacity of a H-BW user may not be fully utilized. This is because, since every neighbor seems identical, a H-BW user may select to provide uploads to several L-BW users who cannot download fast. Hence, we can no longer use Eq. (3) (with $R_{\text{up}H} = C_{\text{up}H}$) to compute U_{HH}. Instead, we need to find a new relation for U_{HH}. This is given in the following lemma:

Lemma 6.

$$U_{\text{HH}} = \sum_{w=0}^{5} R_{\text{HH}}(w) \text{Prob}\{\textit{upload to } w \text{ L-BW } \textit{neighbors}\}, \tag{14}$$

where:

$$R_{\text{HH}}(w) = \begin{cases} \dfrac{C_{\text{up}H} - w U_{\text{HL}}}{5 - w} & \text{if } w < 5, \\ 0 & \text{otherwise} \end{cases}$$

and:

$$\text{Prob}\{\textit{upload to } w \text{ L-BW } \textit{neighbors}\} = \text{Binomial}(5, \alpha, w).$$

Proof. The average rate by which a H-BW user is uploading to a L-BW user is U_{HL}. If a H-BW user is uploading to w L-BW users, then the average upload rate to each H-BW user is equal to $R_{\text{HH}}(w) = \frac{C_{\text{up}H} - w U_{\text{HL}}}{5-w}$, where $C_{\text{up}H} - w U_{\text{HL}}$ is the spare upload capacity of the H-BW user. Further, a H-BW user randomly selects five neighbors to whom to provide uploads because users always have tokens. Hence, Prob{upload to$w$ L-BW neighbors} $=$ Binomial$(5, \alpha, w)$. $\square$

Notice that the way we have related $n^u_{\mathrm{HH}}, n^u_{\mathrm{HL}}, n^u_{\mathrm{LH}}, n^u_{\mathrm{LL}}, U_{\mathrm{HH}}, U_{\mathrm{HL}}, U_{\mathrm{LH}}, U_{\mathrm{LL}}$ to $n^d_{\mathrm{HH}}, n^d_{\mathrm{HL}}, n^d_{\mathrm{LH}}, n^d_{\mathrm{LL}}, D_{\mathrm{HH}}, D_{\mathrm{HL}}, D_{\mathrm{LH}}, D_{\mathrm{LL}}$ in the original system is also valid in the token-based system. Therefore, we can now compute the average download rates using Eqs. (1) and (2), and hence the average download delays using Eqs. (11) and (12), as we did before.

Note that in order to check whether condition $K_{\mathrm{up}} U_{\mathrm{LH}} \geq K_{\mathrm{down}} U_{\mathrm{HL}}$ is satisfied, after solving the system of equations under this assumption, as described above, we then check to see if the resulting U_{HL} satisfies this condition ($K_{\mathrm{up}} U_{\mathrm{LH}} \geq K_{\mathrm{down}} U_{\mathrm{HL}}$). If the condition is not satisfied, we need to find new expressions for U_{HL}, U_{HH}, and n^u_{HL} and resolve the system. This is because, if the condition is not satisfied, it means that L-BW users will not have sufficient tokens to download from H-BW users, and therefore, H-BW users will rarely pick L-BW users to whom to upload. The resulting relation for U_{HL} in this case is similar to Eq. (8) and it is given in the following lemma:

Lemma 7.

$$U_{\mathrm{HL}} = \min\left(\frac{C_{\mathrm{up}H}}{5}, C_{\mathrm{down}L} - n^d_{\mathrm{LL}} U_{\mathrm{LL}}, \frac{K_{\mathrm{up}} U_{\mathrm{LH}}}{K_{\mathrm{down}}}\right). \tag{15}$$

Proof. First, the token earning rate of a L-BW user from a H-BW user is $K_{\mathrm{up}} U_{\mathrm{LH}}$. Hence, the download rate of a L-BW user from a H-BW user cannot exceed $\frac{K_{\mathrm{up}} U_{\mathrm{LH}}}{K_{\mathrm{down}}}$. (Recall that each user keeps track of the amount of tokens that his/her neighbor possesses.) Now, $C_{\mathrm{down}L} - n^d_{\mathrm{LL}} U_{\mathrm{LL}}$ is the spare download capacity of the L-BW user. Clearly, he/she cannot download at a rate faster than this. Finally, as with the proof of Lemma 2, if the spare capacity of the L-BW user is larger than his/her fair share ($\frac{C_{\mathrm{up}H}}{5}$), the user will be downloading from the H-BW user at an average rate equal to his/her fair share. Combining these facts, gives the result. $\square$

Further, we can compute U_{HH} using Eq. (3), as we did before. To compute n^u_{HL}, suppose there are N users in the system. By observing that in the long run the token earning rate of all L-BW users from H-BW users ($n^u_{\mathrm{LH}} K_{\mathrm{up}} U_{\mathrm{LH}} N \alpha$) equals the token spending rate of all L-BW users to H-BW users ($n^u_{\mathrm{HL}} K_{\mathrm{down}} U_{\mathrm{HL}} N(1-\alpha)$), we can write:

$$n^u_{\mathrm{HL}} = \frac{n^u_{\mathrm{LH}} K_{\mathrm{up}} U_{\mathrm{LH}} \alpha}{K_{\mathrm{down}} U_{\mathrm{HL}} (1-\alpha)}. \tag{16}$$

5. Experiments

5.1. Simulation setup

We use an event-driven BitTorrent simulator developed by [26] for our simulations. The detailed simulator description can be found in [4]. We now summarize several important characteristics of this simulator.

- The simulator assumes the bottleneck link of a connection is either a user's upload link or the user's download link, i.e. the simulator assumes the backbone network has infinite bandwidth.
- The simulator simulates the flow-level queueing delay rather than the packet-level queueing delay, which implies that the simulator assumes all connections traversing a link share the link capacity equally, if they are not bottlenecked elsewhere.
- The simulator does not model packet-level TCP dynamics, such as slow start, self-clocking, and packet loss. In addition, the simulator does not simulate the propagation delay.

Notice that these simplifications do not have significant impacts on the results, as argued in [4]. In addition, we implement the proposed token-based scheme to study its impact on the system performance.

To validate our model, we simulate a flash crowd scenario where 200 leechers join the system within 20 s. Leechers will leave the system as soon as they finish their download. We simulate the system until all leechers depart. Because, as we have mentioned earlier, we are interested in the steady state, to avoid the rampup period at the beginning of the simulation, we randomly assign each user 5% of the blocks of the file. Other simulation settings are: (i) there is only

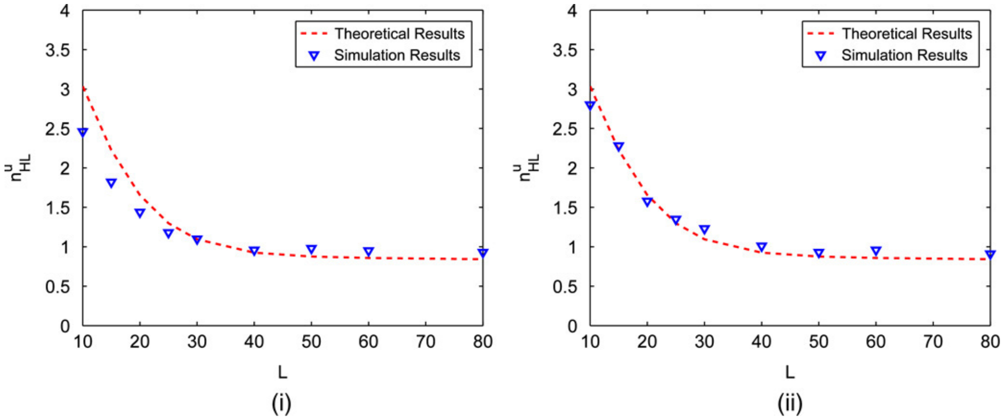


Fig. 3. Average number of L-BW users to whom a H-BW user is uploading: (i) Scenario 1, and (ii) Scenario 2.

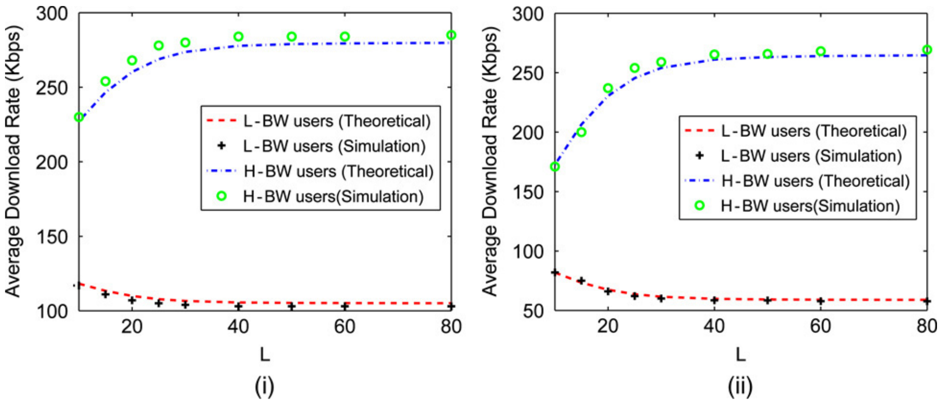


Fig. 4. Average download rate for H-BW and L-BW users: (i) Scenario 1, and (ii) Scenario 2.

one seed in the system and the upload link capacity of the seed is 800 Kbps, (ii) the file size is 300 MB and the block size is 512 KB, and (iii) the maximum number of concurrent upload transfers is five.

We present simulation results for two scenarios, which, as we shall see, yield qualitatively different results when the token- based scheme is used. In both scenarios the percentage of L-BW users is $\alpha = 0.8$, $C_{\mathrm{down}H} = 600$ Kbps, and $C_{\mathrm{up}H} = 300$ Kbps. For Scenario 1 we have $C_{\mathrm{down}L} = 300$ Kbps, $C_{\mathrm{up}L} = 100$ Kbps, and for Scenario 2 we have $C_{\mathrm{down}L} = 150$ Kbps, $C_{\mathrm{up}L} = 50$ Kbps. Note that we have done extensive simulations with different values of the parameters and the results are similar.

5.2. *Model verification*

5.2.1. *Simulation results for the original BitTorrent system*

We first study how n^u_{HL}, the average number of L-BW users that are downloading from a H-BW user, behaves as the number of neighbors L increases. This will give us intuition later on, when we show how the download rates and delays change as a function of L. Both theoretical and simulation results are shown in Fig. 3.

First, we observe from the plots that Eq. (9) can correctly predict n^u_{HL} for both cases. Further, we can observe that n^u_{HL} decreases as L increases. This is because when L is small H-BW users cannot find enough H-BW peers to upload to, and thus they have to whom to provide uploads more L-BW users. As L increases there are more H-BW users to whom to upload, and thus there is no need to upload to L-BW users.

The download rates for both H-BW and L-BW users with respect to L are shown in Fig. 4. Notice that the results correspond to the period $(t_1, t_2]$ in Fig. 2. (Recall that in the interval $(t_3, t_4]$ the download rate of L-BW users is just equal to their upload capacity.) Again, we can observe from the plots that our mathematical model is quite accurate.

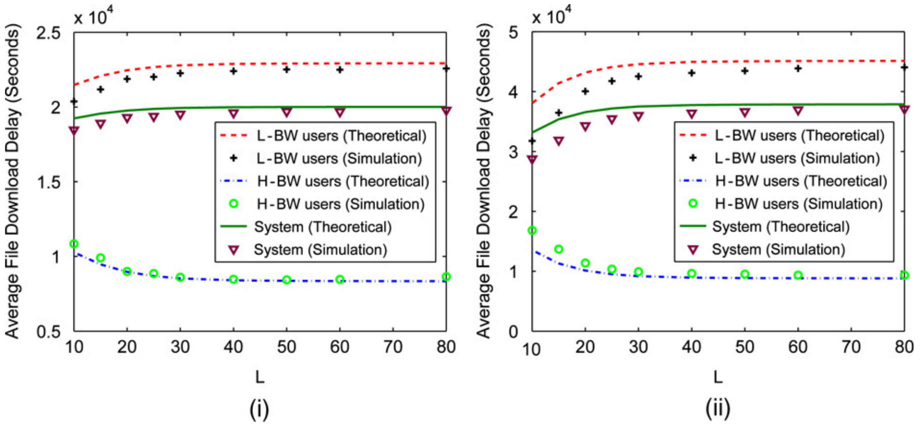


Fig. 5. Average file download delay for H-BW users, L-BW users, and for the system: (i) Scenario 1, and (ii) Scenario 2.

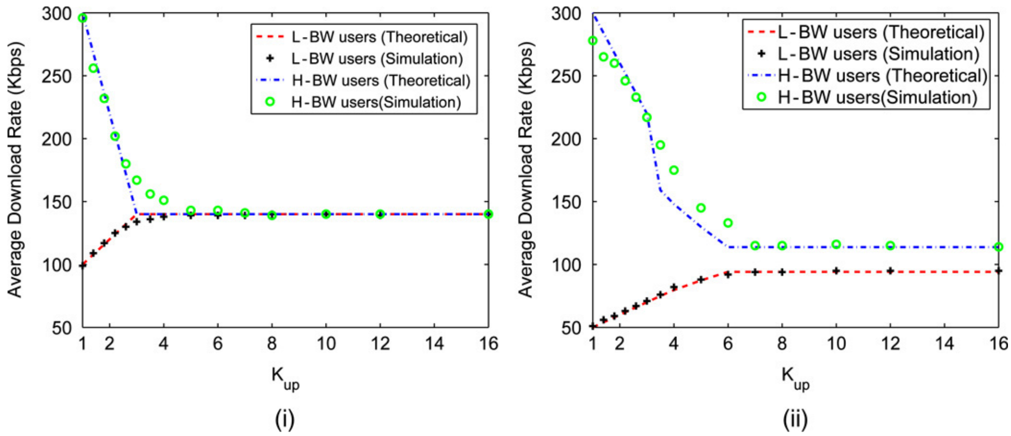


Fig. 6. Average download rate for H-BW and L-BW users: (i) Scenario 1, and (ii) Scenario 2.

Notice that the download rate of H-BW users increases and the download rate of L-BW users decreases as L increases. This can be explained in a similar manner as with n^u_{HL}. As L increases H-BW users provide uploads to fewer L-BW users and to more H-BW users.

Finally, theoretical and simulation results for the average file download delay are shown in Fig. 5. We can observe from the plots that our model can correctly predict the average file download delay for H-BW users, L-BW users, and for the whole system.[9]

5.2.2. Simulation results for the token-based system

We now let $K_{\mathrm{down}} = 1$ and study how the token-based system behaves for different values of K_{up}, for the scenarios we have considered earlier. We fix $L = 40$, which is a typical value in BitTorrent [3]. Fig. 6 shows the theoretical and simulation results for the download rate of H-BW and L-BW users.

First, from the plots, we see again that theoretical and simulation results match. Further, we make the following interesting observation: the download rate of H-BW users decreases and the download rate of L-BW users increases as K_{up} increases. This is because as K_{up} increases L-BW users earn tokens at a faster rate and they can download more data from H-BW users. This, however, means that H-BW users provide fewer uploads to other H-BW users. Thus, H-BW users have to download now from more L-BW users, and hence their download rate decreases. Further, it is interesting to point out that in the first scenario the two classes of user have the same download rate for large

[9] Note that the results in Fig. 5 correspond to the whole time interval $(t_0, t_4]$ in Fig. 2. The relatively small discrepancies are due to the fact that we do not model the transient periods ($(t_0, t_1]$ and $(t_2, t_3]$), as mentioned before.

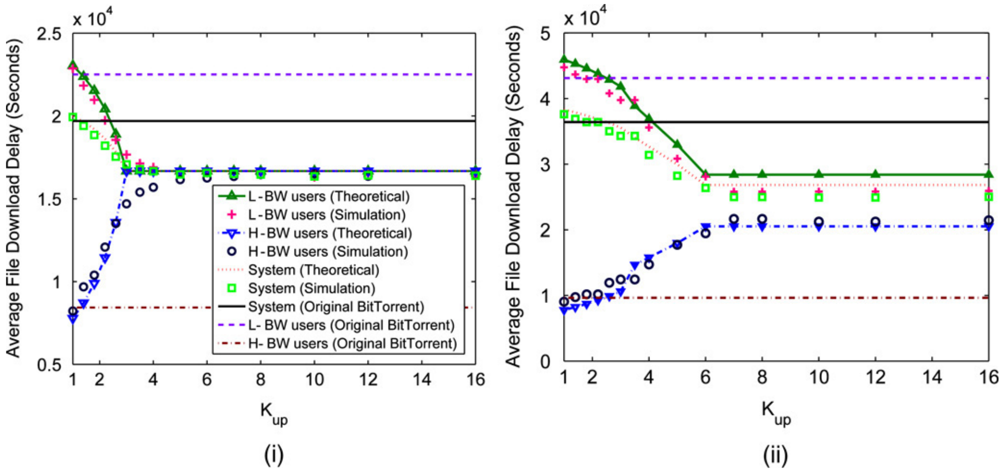


Fig. 7. Average file download delay for H-BW users, L-BW users, and for the system: (i) Scenario 1, and (ii) Scenario 2.

K_{up}, whereas in the second scenario the download rates of the two classes are never equal. This is because in the first scenario (for large K_{up}) both classes of user are downloading from a similar number of H-BW users, and since $C_{\text{down}L} = C_{\text{up}H}$ and $C_{\text{down}H} > C_{\text{up}H}$, both classes of user can fully utilize the H-BW users' upload capacity. In contrast, in the second scenario, while both classes of user are downloading from a similar number of H-BW users (for large K_{up}), since $C_{\text{down}H} > C_{\text{up}H}$ but $C_{\text{down}L} < C_{\text{up}H}$, only H-BW users can fully utilize the upload capacity of other H-BW users from whom they are downloading.

Fig. 7 shows theoretical and simulation results for the average file download delay for H-BW users, L-BW users, and for the whole system. For comparison, the plots also show the corresponding average download delay in the original BitTorrent system.

As before, we observe that our model predicts the simulation results quite accurately. Further, we observe that when $K_{\text{up}} = 1 = K_{\text{down}}$, the performance of the token-based system is almost identical to that of the original BitTorrent system. However, as K_{up} increases, the overall system performance can be improved compared to the original BitTorrent system.[10] This is because in the token-based system L-BW users are downloading from more H-BW users if K_{up} is large, since as we have mentioned earlier, L-BW users can gain tokens fast. However, as mentioned earlier, we are sacrificing the perceived performance of H-BW users. This motivates us to quantify next how "unfair" the token-based scheme becomes to H-BW users as K_{up} increases.

5.3. Impact of the proposed token-based scheme on fairness

To quantify "fairness" we use the upload-to-download ratio of a user, which is defined as the user's upload rate divided by his/her download rate.[11] Fig. 8 shows how the upload-to-download ratio behaves as we vary K_{up}, for each class of user.

From these plots we observe that the upload-to-download ratio is almost the same for both classes of user when $K_{\text{up}} = 1 = K_{\text{down}}$. This implies that the system is fair. However, as K_{up} increases, the corresponding ratio for H-BW users increases and for L-BW decreases, as expected. (This suggests that the system becomes unfair.)

Looking at Figs. 7 and 8 we can conclude that we can tradeoff between overall system performance and fairness. Using our analytical model we can predict how much "fairness" we are sacrificing and what performance is achieved. For example, one can enforce fairness by setting $K_{\text{up}} = 1 = K_{\text{down}}$, or can minimize the system's average download delay by choosing a large value for K_{up}. Further, one can also operate somewhere between these two extremes by setting the appropriate value for K_{up}.

[10] Note that one can achieve more significant performance improvements, depending on the values of $C_{\text{down}H}$, $C_{\text{down}L}$, $C_{\text{up}H}$, and $C_{\text{up}L}$ than those we show here [27].

[11] This metric has been also used to quantify fairness in other studies as well, *e.g.* [4,18].

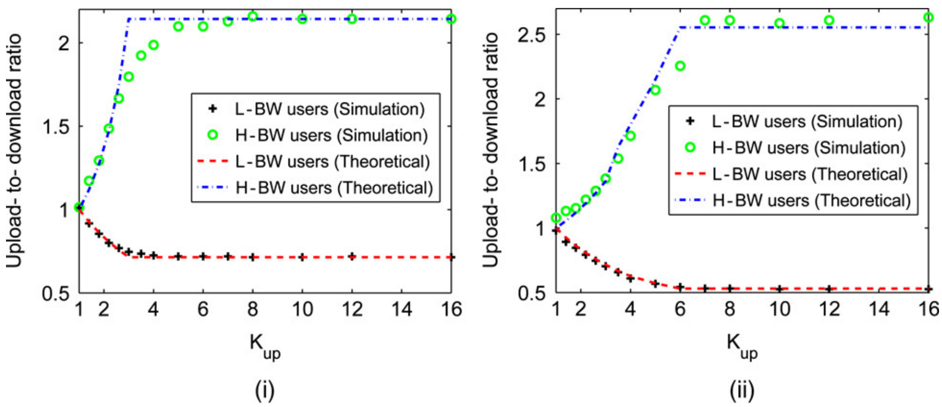


Fig. 8. Upload-to-download ratio for H-BW and L-BW users: (i) scenario 1, and (ii) scenario 2.

6. Conclusion and future work

In this paper we have proposed a mathematical model to study the performance of heterogeneous BitTorrent-like systems. In particular, we have presented a model that can be used to predict the average file download delay among users with different capacities. Further, we have proposed a flexible token- based TFT scheme that can be used to tradeoff between fairness and system performance. We have extended our mathematical model in order to predict the system performance under the proposed scheme and for tuning the scheme's parameters. Our results have been verified using extensive simulations.

We have several interesting directions for future work. First, we plan to use our model, thoroughly to study the system's performance when one varies the system's parameters (*e.g.*, when one varies L, the number of neighbors returned by the tracker, when one varies the maximum number of upload connections a user can provide, etc.). In particular, we are interested in investigating if there are combinations for the values of the system's parameters that achieve optimal download delays.

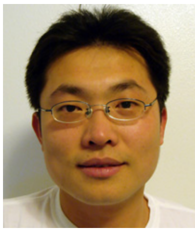
Second, we plan to extend our analysis to study the system's dynamics, and transient behavior. In particular, we want to extend our model to be able to predict how the download delays behave when a file has just become popular, and hence the number of users in the system has not reached its steady state value yet, as well as the download delays while existing users depart the system.

Finally, we plan to perform experiments on PlanetLab [28] to verify our model under more realistic settings.

References

[1] S. Saroiu, K.P. Gummadi, R.J. Dunn, S. Gribble, H.M. Levy, An analysis of internet content delivery systems, in: Proc. of the Fifth Symposium on Operating System Design and Implementation, OSDI, 2002.
[2] A. Parker, The true picture of peer-to-peer filesharing. URL: http://www.cachelogic.com/, Accessed February 2007.
[3] Bittorrent. URL: http://www.bittorrent.com/protocol.html, Accessed February 2007.
[4] A. Bharambe, C. Herley, V. Padmanabhan, Analyzing and improving bittorrent performance, in: Proc. of IEEE INFOCOM, 2006.
[5] D. Qiu, R. Srikant, Modeling and performance analysis of bittorrent-like peer-to-peer networks, in: Proc. of ACM SIGCOMM, 2004.
[6] F. Piccolo, G. Neglia, The effect of heterogeneous link capacities in bittorrent-like file sharing systems, in: Proc. of 1st International Workshop on Hot Topics in Peer-to-Peer Systems, Hot-P2P '04, 2004.
[7] J. Wang, C. Yeo, V. Prabhakaran, K. Ramchandran, On the role of helpers in peer-to-peer file download systems: Design, analysis, and simulation, in: Proc. of 6th International Workshop on Peer-to-Peer Systems, IPTPS '07, 2007.
[8] B. Fan, D.-M. Chiu, J. Lui, Stochastic differential equation approach to model bittorrent-like file sharing systems, in: Proc. of 14th IEEE International Workshop on Quality of Service, 2006.
[9] F. Clevenot, P. Nain, K. Ross, Multiclass p2p networks: Static resource allocation for service differentiation and bandwidth diversity, in: Proc. of 24th IFIP WG 7.3 International Symposium on Computer Performance, Modeling, Measurements and Evaluation, 2005.
[10] 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 (2007).
[11] Y. Tian, D. Wu, K. Ng, Modeling, analysis and improvement for bittorrent-like file sharing networks, in: Proc. of IEEE INFOCOM, 2006.
[12] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, X. Zhang, Measurements, analysis, and modeling of bittorrent-like systems, in: Proc. of Internet Measurement Conference, 2005.

[13] N. Liogkas, R. Nelson, E. Kohler, L. Zhang, Exploiting bittorrent for fun (but not profit), in: Proc. of 5th International Workshop on Peer-to-Peer Systems, IPTPS '06, 2006.
[14] D. Hales, S. Patarin, How to cheat bittorrent and why nobody does, Tech. Rep. UBLCS-2005-12, University of Bologna, Italy, 2005.
[15] Bigchampagne. URL: http://www.bigchampagne.com/, Accessed February 2007.
[16] J. Pouwelse, P. Garbacki, D. Epema, H. Sips, The bittorrent p2p file-sharing system: Measurements and analysis, in: Proc. of 4th International Workshop on Peer-to-Peer Systems, IPTPS '05, 2005.
[17] M. Izal, G. Keller, E. Biersack, P. Felber, A. Hamra, L. Erice, Dissecting bittorrent: Five months in a torrent's lifetime, in: Proc. of Passive and Active Measurements, 2004.
[18] R. Thommes, M. Coates, Bittorrent fairness: Analysis and improvements, in: Proc. Workshop Internet, Telecom. and Signal Proc., December 2005.
[19] W.-C. Liao, F. Papadopoulos, K. Psounis, An efficient algorithm for resource sharing in peer-to-peer networks, in: Proc. of IFIP Networking, 2006.
[20] W.-C. Liao, F. Papadopoulos, K. Psounis, A peer-to-peer cooperation enhancement scheme and its performance analysis, Journal of Communications 1 (2006).
[21] B. Cohen, Incentives build robustness in bittorrent, in: Workshop on Economics of Peer-to-Peer Systems, 2003.
[22] Z. Ge, D.R. Fiegueiredo, S. Jaiswal, J. Kurose, D. Towsley, Modeling peer-peer file sharing systems, in: Proc. of IEEE INFOCOM, 2003.
[23] X. Yang, G.D. Veciana, Service capacity of peer to peer networks, in: Proc. of IEEE INFOCOM, 2004.
[24] R. Gaeta, M. Gribaudo, D. Manini, M. Sereno, Analysis of resource transfers in peer-to-peer file sharing applications using fluid models, Performance Evaluation 63 (2006) 149–174.
[25] S. Saroiu, K. Gummadi, R. Dunn, S. Gribble, A measurement study of peer-to-peer file sharing systems, in: Proc. of Multimedia Computing and Networking 2002, MMCN '02, 2002.
[26] A. Bharambe, C. Herley, V. Padmanabhan, Microsoft research simulator for the bittorrent protocol. URL: http://www.research.microsoft.com/projects/btsim, Accessed February 2007.
[27] W.-C. Liao, F. Papadopoulos, K. Psounis, Performance analysis of bittorrent-like systems in heterogeneous networks, Tech. Rep. CENG-2007-2, University of Southern California, 2007.
[28] Planetlab. URL: http://www.planet-lab.org/, Accessed February 2007.



Wei-Cherng Liao is a Ph.D. candidate in Electrical Engineering at the University of Southern California. He received his first degree and the M.Sc. degree from the department of Electrical and Computer Engineering of National Taiwan University, Taiwan, in June 1998 and June 2000 respectively. His research interests include modeling, and performance prediction/analysis of computer networks and the incentive scheme of P2P system.

Fragkiskos Papadopoulos is a Ph.D. candidate in Electrical Engineering at the University of Southern California. He received his first degree from the department of Electrical and Computer Engineering of the National Technical University of Athens, Greece, in June 2002 and the M.Sc. degree in Electrical Engineering, specializing in computer networks, from the University of Southern California in May 2004. His research interests include modeling, simulation, and performance prediction/analysis of computer networks and P2P systems. He is a recipient of the Fulbright Scholarship.

Konstantinos Psounis is an assistant professor of Electrical Engineering and Computer Science at the University of Southern California. He received his first degree from the department of Electrical and Computer Engineering of National Technical University of Athens, Greece, in June 1997, the M.S. degree in Electrical Engineering from Stanford University, California, in January 1999, and the Ph.D. degree in Electrical Engineering from Stanford University in December 2002.

Konstantinos models and analyzes the performance of computer networks, sensor and mobile systems, and the Web. He also designs methods and algorithms to solve problems related to such systems. He is the author of more than 30 research papers on these topics. Konstantinos has received faculty awards from NSF and the Zumberge foundation, has been a Stanford graduate fellow throughout his graduate studies, and has received the best-student National Technical University of Athens award for graduating first in his class.