The Delicate Tradeoffs in BitTorrent-like File Sharing Protocol Design

Bin Fan
Dept. of Computer Science & Eng.
The Chinese University of Hong Kong
Email: bfan@cse.cuhk.edu.hk

Dah-Ming Chiu
Dept. of Information Eng.
The Chinese University of Hong Kong
Email: dmchiu@ie.cuhk.edu.hk

John C.S. Lui
Dept. of Computer Science & Eng.
The Chinese University of Hong Kong
Email: cslui@cse.cuhk.edu.hk

***Abstract*— The BitTorrent (BT) file sharing protocol is popular due to its scalability property and the incentive mechanism to reduce free-riding. However, in designing such P2P file sharing protocols, there is a fundamental "*tussle*" between keeping peers, specially the more resourceful ones, in the system for as long as possible to help the system achieve better performance and allowing peers finish their download as quickly as possible. The current BT protocol represents only "*one*" possible implementation in this whole design spectrum. In this paper, we characterize the "*complete*" design space of BT-like protocols. We use fairness index to measure the fairness that incorporates the contribution peers make. We show that there is a wide range of design choices, ranging from optimizing the performance of file download, to optimizing the fairness measure. More importantly, we show that there is a simple and easily implementable design knob which can be used to choose a particular operating point in the design space. We then discuss different algorithms (centralized versus distributed) in realizing the design knob. We also carry out performance evaluation to quantify the merits and properties of the BT-like file sharing protocols.**

I. Introduction

A class of peer-to-peer (P2P) content distribution protocols, typified by a P2P file sharing application called BitTorrent (BT) [6], is catching the attention of computer networking research community. Unlike the traditional client/server style of content distribution in which performance usually degrades as the number of clients increases, in idealized scenarios BT system's performance may actually improve when the number of participants increases. The key idea that makes BT scalable is credited to its *cooperative* mechanism. It can be explained intuitively as follows. The server's content is partitioned into many small pieces. Each peer can get the content either from the server, or from other peers holding those pieces it does not already have. Each peer offers upload service to other peers only to the extent the service is reciprocated. Also, each peer tries to obtain the rarest piece so as to maximize its ability to serve others hence also the service it will receive. By linking the service each peer can receive to its contribution to others, BT protocol successfully makes each peer play a role of a server and a client at the same time. Therefore, as the number of peers increases, the service capacity of the whole system increases as well.

Although the performance scales well with the peer population, such kind of protocol may face the following throughput-fairness dilemma in design: Peers that participate in a BT file sharing process are likely heterogeneous. That is, they likely have different uploading/downloading capacities. A *fat* node, connected to the Internet via the latest broadband access technology, typically has a high capacity for uploading; on the other hand, a *thin* node is one that has a lower uploading capacity. Therefore, the system throughput of a BT-like file sharing protocol also depends critically on how long the fat peers are retained in the system so as to sustain as large an aggregated system capacity as possible.

However, keeping the fat peers in the system to serve others seems inherently unfair. The upload capacity is usually a parameter controllable by the peer. If by offering a high uploading rate leads to receive lower service, a user's rational choice would be to artificially lower its upload capacity. Intuitively, a user would look for service differentiation proportional to the amount of contribution it makes (in BT system, contribution refers to uploading). The more a protocol offers such service differentiation, the more *incentive compatible* it is, and hence likely to succeed.

In this paper, we analytically explore this fundamental tradeoff by defining two separate design objectives:

1) *Performance*: In a BT-like P2P file sharing system, the basic performance metric is to minimize the *average downloading time*. Naturally, this is compatible with each peer's wish, since they all want to finish downloading as soon as possible.
2) *Fairness*: To make the system scalable, peers need to contribute (via uploading) in order to receive service. A well-designed protocol should encourage peers to contribute in an incentive compatible way: those who contribute more should receive a better service (i.e., achieve a smaller file downloading time) than those who contribute less.

In this paper, we formally define these (*performance* and *fairness*) metrics, and use them to analyze different outcomes of BT-like file sharing algorithms with heterogeneous peers. In particular, we address the following important questions:

- *If an algorithm optimized performance, what would happen to fairness?*

1-4244-0593-9/06/$20.00 ©2006 IEEE

- *If fairness is maximized, what happens to performance?*
- *To what degree of performance and fairness the current BT file sharing protocol is achieving?*
- *Can we characterize the different tradeoffs between performance and fairness, and what does it mean to implement each of these tradeoff points?*

Our study is based on a simple model of peers sharing each others' upload capacity. Peers are assumed to randomly arrive, and join the system, and leave after completing downloading. For the general case of multiple types of peers (in terms of upload/download capacity), the complete spectrum of performance versus fairness tradeoffs is analyzed.

The rest of the paper is organized as follows: In Section II, we present a mathematical model and define the performance measures which help us illustrate the whole design state space. In Section III, we describe several rate assignment strategies to achieve different forms of optimality, namely, (a) optimal downloading time, (b) optimal fairness and (c) max-min fairness. In Section IV, we present a distributed framework via uploading policy and show how this can achieve different points in the whole design space. In Section V, we carry out performance evaluation to quantify the merits of our proposed method. Related work is given in Section VI and Section VII concludes.

II. Mathematical Model

In this section, we present a simple mathematical model of BT-like file sharing protocol that allows us to study the tradeoff between performance and service differentiation for peers with heterogeneous bandwidth capacity.

A. The Generic Model of Uplink Sharing

First we describe a generic model of a BT-like file sharing system. The system is designed to disseminate a specific file of finite size among the set of peers $\mathcal{N} = \{1, 2, \ldots, n\}$. Without loss of generality, we assume the file size to be unity. Each peer i in this file sharing session has its upload capacity U_i and download capacity D_i. In other words, U_i (D_i) is the maximum uploading (downloading) rate that peer i can achieve. We also assume the following constraint

$$U_i \leq D_i, \tag{1}$$

which is true for most of the recent Internet access technologies and verified by the measurement study in [19]. Assume each peer always has a sufficient part of the file of interest so that it can upload that part of the file to one or more peers. Let u_i denote the actual uploading rate of peer i, which has to satisfy the upload capacity constraint:

$$u_i \leq U_i. \tag{2}$$

At the same time, peer i can achieve its downloading rate d_i, which has also to satisfy:

$$d_i \leq D_i. \tag{3}$$

From the system's perspective, the total downloading rates of all peers must balance out with the total uploading rates:

$$\sum_i d_i = \sum_i u_i. \tag{4}$$

This model is similar to the *uplink capacity sharing problem* proposed in [15], in which the network is assumed not to be the bottleneck, and the limit to system throughput is the uplink capacities. Since there are $|\mathcal{N}| = n$ peers in the system, any rate vector $u = [u_1, u_2, \ldots, u_n]$ and $d = [d_1, d_2, \ldots, d_n]$ that satisfy the above constraints of (2), (3) and (4), is considered as a *feasible strategy* of uplink sharing. The performance of a P2P file sharing system can then be evaluated based on the feasible allocation strategy of the uplink capacity resource [4], [15].

Note that the uplink capacity sharing model involves a perfect scheduling assumption. In reality, there is always some probability that the content a peer has is not of interest to other peers, so this peer cannot offer its upload service to other peers. If there are many peers in a file sharing system and the wanted file is divided into a large number of chunks, it is likely that each peer is able to obtain some distinct content to serve other peers (i.e., by following the *rarest first* heuristic in BT).

B. A Dynamic Model of Multiple Classes of Peers

In our paper, we extend the generic model in the following aspects:

- In [4], [15], all peers join the system at the beginning and leave the system together after the last peer finishes its downloading process. In this paper, we allow dynamic arrival and leaving of peers. More specifically, peers arrive to the system with an average rate λ. Once a peer is in the system, it performs downloading and uploading at the same time. Peers do not abort and remain in the system until they finish the file download. We assume upon completion of the file download, a peer will leave the system immediately. Note that in the BT protocol, there is no incentive to become a seeder after downloading the file.
- In the dynamic version of the model, the number of peers in the system varies with time. Therefore, instead of assuming n peers in the system, we assume there are n classes of peers in the system. For each new peer arrival, with probability p_i, it is assumed to be of type i. Thus the average arrival rate of type i is λp_i, with $\sum_{i=1}^{n} p_i = 1$. For a type i peer, its upload (download) capacity is U_i (D_i). For simplicity of presentation, we assume $U_1 > U_2 > \ldots > U_n$. Assume all peers of the same type get the same uploading rate $u_i(t)$ and downloading rate $d_i(t)$ respectively at time t. For all type-i peers, according to (2) and (3)we have $u_i(t) \leq U_i$ and $d_i(t) \leq D_i$. Since type-1 peers has the highest uploading capacity, for simplicity, in the following we may mention

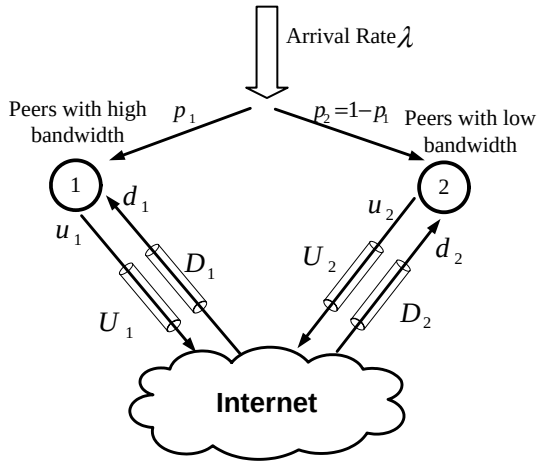

Fig. 1. The System Model

type-1 peers as *"fat"* peers, and all the other types of peers as *"thin"* peers.

Figure 1 illustrates the dynamic model of the system when $n = 2$ and associated notations. Here we have two classes of peers: Type-1 ("fat") peers with higher upload capacity and type-2 ("thin") peers which have lower upload capacity.

C. Performance Metric

For traditional network services such as FTP and email, the service capacity is determined by the dedicated server (assuming that the network has sufficient resource). As a result, when the number of request increases, the bottleneck is on the server side and the quality of service deteriorates rapidly. BT-like file sharing system resolves this problem by distributing the service to every participating peers, which serve both as resource provider and consumer. Let $N_i(t)$ denote the number of type-i peers in the system at time t. The file uploading service capacity of the entire system, which is denoted as $\mathcal{C}(t)$, can be expressed as:

$$\mathcal{C}(t) = \sum_{i=1}^{n} u_i(t) N_i(t). \tag{5}$$

Here $u_i(t)N_i(t)$ is the uploading capacity brought by type-i peers. From (5), one can find that the service capacity is also related to the number of different peers and the distribution of peers' uploading capacity.

Let us consider the steady state behavior of the system. We define the average number of type-i peers in the steady state as N_i. One can use the Little's result to relate the number of type-i peers and their average file downloading time as:

$$N_i = (p_i \lambda) \cdot T_i = \frac{p_i \lambda}{d_i} \quad \text{for } i \in \{1, \ldots, n\}. \tag{6}$$

Here, T_i and d_i are the average downloading time and downloading rate for type-i peer. Substitute (6) into (5) for $i = \{1, \ldots, n\}$, the steady state system service capacity $\mathcal{C}$ is:

$$\mathcal{C} = p_1 \lambda \frac{u_1}{d_1} + \ldots + p_n \lambda \frac{u_n}{d_n}. \tag{7}$$

In the steady state, the system service capacity $\mathcal{C}$ should be equal to the total arrival rate λ (since the size of the file is 1), we get the following equation:

$$p_1 \frac{u_1}{d_1} + \ldots + p_n \frac{u_n}{d_n} = 1. \tag{8}$$

Let $c_i = \frac{u_i}{d_i}$. In BT terminology, c_i is called *"share ratio"* which represents the amount of uploading divided by the amount of downloading. A share ratio of 1.0 means that a peer has uploaded as much data as it has downloaded. A share ratio greater than 1 means that a peer has uploaded more than it has downloaded. Naturally a peer with a higher share ratio makes better contribution to the system. Now (8) becomes:

$$p_1 c_1 + \ldots + p_n c_n = 1, \tag{9}$$

which means in a steady and balanced system, the average share ratio of all types of peers should be 1.

Also, applying the Little's result, the average system downloading time for all peers can be expressed as:

$$T = \frac{N_1 + \ldots + N_n}{\lambda} = \frac{p_1}{d_1} + \ldots + \frac{p_n}{d_n}. \tag{10}$$

The above two equations give the feasible solution space, or operating points for the P2P file sharing algorithm, as well as the resultant performance in terms of the average downloading time T. The feasible solution space is characterized in terms of the uploading and downloading rates of the two types of peers, $\{u_i, d_i | i = 1, \ldots n\}$.

So far, we have derived the expression for a particular metric of interest, namely, average system downloading time T. Let's turn our focus to the other system metric of interest, namely, fairness, in the next subsection.

D. Fairness Metric

The issue of *fairness* has long been studied in the networking community. For example, in [18], [20] one can find various expressions to quantify the fairness among competing entities. Now we will discuss the fairness issue in the context of BT-like P2P file sharing. Note that share ratio is generally considered as a good indicator to measure the contribution to the system with respect to the obtained service. Share ratios are more important on BitTorrent community than they are on other peer-to-peer file sharing networks, because many BT trackers require peers to maintain a minimum share ratio, say 0.5. Due to the physical meaning of share ratio, a well-designed system should try to make all its peers maintain the same share ratio. So in this paper, we want to measure the fairness of a system according to the peers' share ratios, by Fairness Index [18]:

$$\mathcal{F} = \frac{(\sum_{i=1}^{n} x_i)^2}{n(\sum_{i=1}^{n} x_i^2)} \tag{11}$$

Equation (11) measures how equal $x_1, \ldots, x_2$ are. Assume after reaching steady state, all together K peers have finished downloading, and K_i is the number of type-i peers who finished downloading. Therefore, $K_i = p_i \cdot K$. Substitute $x_i = c_i$ and use (9), we have:

$$\begin{aligned}\mathcal{F} &= \frac{(K_1c_1 + \ldots + K_nc_n)^2}{K \cdot (K_1c_1^2 + \ldots + K_nc_n^2)} = \frac{(p_1c_1 + \ldots + p_nc_n)^2}{p_1c_1^2 + \ldots + p_nc_n^2} \\ &= \frac{1}{p_1c_1^2 + \ldots + p_nc_n^2}. \end{aligned} \tag{12}$$

It is important to point out that from the properties of Fairness Index we know that when $c_1 = \ldots = c_n$, $\mathcal{F}$ reaches its maximum.

So far we have defined two metrics for the P2P uplink sharing problem, performance (average delay T) and fairness, both expressed in terms the feasible upload and download rate assignments vector $[u_1, \ldots, u_n]$ and $[d_1, \ldots, d_n]$. The different solutions to the P2P uplink sharing problem, hence the design of the P2P file sharing algorithm, can thus be understood based on the merit of these assignments.

III. Rate Assignment Strategies

A. Uploading Rate

Since peers heterogeneous and autonomous (they can choose different uploading and downloading rates), it is important for us to seek the fundamental understanding of the proper rate assignment of u_i and d_i in the whole design space of BT-like protocols so that desirable tradeoff can be achieved.

First let us investigate the rate assignment for peers' uplink. BitTorrent protocol is generally considered very effective in content distribution. In [2], experimental result shows that BitTorrent performs near-optimally in terms of uplink bandwidth utilization. So for simplicity we assume

$$u_i = U_i, \tag{13}$$

which implies that in the design of BT protocol, the upload capacity of type-i peers is always saturated.

In the following, we want to derive the feasible settings of d_i when the system (a) minimizes the average downloading time T, or (b) maximizes the fairness measure $\mathcal{F}$, or (c) achieves the max-min allocation of the downloading rate.

B. Rate Assignment for Optimal Downloading Time

First, we investigate the condition to achieve the optimal average system downloading time T. To minimize the average system downloading time T defined by (10), one needs to solve the following constrained optimization problem:

$$\begin{aligned} \text{Min} \quad & T = \frac{p_1}{d_1} + \ldots + \frac{p_n}{d_n} \\ \text{s.t.} \quad & p_1\frac{U_1}{d_1} + \ldots + p_n\frac{U_n}{d_n} = 1, \\ & 0 \le d_i \le D_i, i = 1, \ldots, n \end{aligned} \tag{14}$$

The first equality constraint is the steady state condition given by (8). The other inequality constraints are due to the physical capacity: D_i and U_i keep the relationship assumed in Section II.

To find the optimal solutions for d_i, we can use the method of Lagrangian multiplier. Define

$$L(\mathbf{d}, \boldsymbol{\lambda}, \nu) = \sum_{i=1}^{n} \frac{p_i}{d_i} + \sum_{i=1}^{n} \lambda_i(d_i - D_i) + \nu(\sum_{i=1}^{n} p_i \frac{U_i}{d_i} - 1).$$

The Karush-Kuhn-Tucker (KKT) conditions [3] for the above optimization are:

$$\begin{aligned} &\frac{\partial L}{\partial d_i} = 0,\ 0 \le d_i \le D_i,\ \lambda_i(d_i - D_i) = 0, \quad i = 1, \ldots, n \\ &p_1\frac{U_1}{d_1} + \ldots + p_n\frac{U_n}{d_n} = 1;\ \ \boldsymbol{\lambda} \succeq 0. \end{aligned}$$

Solving this optimization problem, one can determine the proper choice of uploading and downloading rates as:

$$\begin{aligned} \text{Type-1 peer :} \quad & d_1 = \frac{p_1U_1}{1 - \sum_{i=2}^{n} p_i \frac{U_i}{D_i}}, \\ \text{Type-i peer :} \quad & d_i = D_i, \quad i = 2, \ldots, n. \end{aligned} \tag{15}$$

The rate assignment strategy to achieve the optimal downloading time by (15) gives us the following insights:

- Since $d_i = D_i$ for $i = 2, \ldots, n$, in order to achieve optimal average downloading time, the system will provide the "thin"(other than type-1) peers as much uploading resource as possible so that these peers can fully utilize their downlink capacity. From the inequality $U_i < D_i$ for $i = 2, \ldots, n$, one can observe that the "thin" peers get more than they contribute to the system. When $u_1 = U_1 > d_1$, one can observe that although the "fat"(type-1) peers have better access technology to the Internet than the "thin" peers, they can only download the file at the rate which is *less than* their uploading rate. Under this rate assignment strategy, the "fat" peers are actually helping the "thin" peers and this is why the system can achieve the optimal performance of file downloading time.
- Consider the service difference of uploading resource for both "fat" and "thin" peers, we have:

$$d_1 - d_k = \frac{p_1}{1 - \sum_{i=2}^{n} p_i \frac{U_i}{D_i}} U_1 - D_k. \tag{16}$$

  It is possible for the "thin" peers to receive *better* service than those "fat" peers, even when "fat" peers are contributing more to the system! It indicates the *cost* of achieving the optimal file downloading time is in terms of fairness.

Under this policy, the achieved performance measure is

$$T_{opt} = \frac{1}{U_1} + \sum_{i=2}^{n} \frac{p_i}{D_i} \frac{U_1 - U_i}{U_1},$$

and the the fairness measure is

$$\mathcal{F}_{opt} = \frac{1}{p_1(\frac{1-\sum_{i=2}^{n} p_i U_i/D_i}{p_1})^2 + \sum_{i=2}^{n} p_i(\frac{U_i}{D_i})^2}.$$

C. Rate Assignment for Optimal Fairness

Now let's derive the strategy to achieve the optimal fairness. From the properties of Fairness Index, we know that, the condition for $\mathcal{F}$ to be 1 is:

$$c_1 = \ldots = c_n = \gamma.$$

Due to the steady state condition of (9), we have:

$$\gamma(p_1 + \ldots + p_n) = 1.$$

Since $p_1 + \ldots + p_n = 1$ we get $\gamma = 1$, indicating $d_i = u_i$ for all type i. This implies that if we want to achieve perfect fairness of $\mathcal{F} = 1$, the system needs to ensure that each type of peers can only receive as much resource as it contributes.

We can solve the above optimization problem using a similar technique as described before. Solving this optimization problem, we obtain:

$$\text{Type-i peer:} \qquad d_i = u_i = U_i. \tag{17}$$

The rate assignment strategy given in (17) indicates us that, to achieve optimal fairness, the system just needs to allocate the downloading rate to each peer equivalent to its contribution. One can observe this service differentiation is fair:

- Due to $U_i < D_i$, all peers wouldn't saturated their downlink capacity.
- The share ratio is 1 for all peers. A "fat" peer can download faster than a "thin" peer since this "fat" peer contributes more. So the peers have some incentive to contribute more.

In summary, the average downloading time for this policy is:

$$T_{fair} = \frac{p_1}{U_1} + \ldots + \frac{p_n}{U_n},$$

and the fairness metric is:

$$\mathcal{F}_{fair} = 1.$$

D. Rate Assignment for Max-min Allocation

So far, we explored the conditions to achieve the optimal file downloading time and optimal fairness. Another possibility is for us to consider a rate assignment strategy in which one can achieve the *max-min* allocation index. Let us first state the definition of max-min allocation [17].

Definition *1:* Consider a set $\mathcal{A} \subset R^n$. A vector $\vec{x}$ is "max-min allocation on set $\mathcal{A}$" if and only if

$$(\forall \vec{y} \in \mathcal{A})\,(\exists s \in \{1,\ldots,n\})\; y_s > x_s \implies (\exists t \in \{1,\ldots,n\})\; y_t < x_t \le x_s. \tag{18}$$

In other words, any feasible increase of one component of the max-min allocation, say x_i, will inevitably result in decreasing another component, x_j, which is no bigger than x_i to start with.

The max-min allocation can be constructed using a waterfilling algorithm [11], [12], increasing the downloading rates of all types of peers d_1, d_2 to d_n simultaneously from 0 together. *If none of the downloading rates is a bottleneck*, eventually we will arrive at a point when $d_1 = \ldots = d_n = d$.

The max-min optimization problem is defined as:

$$\begin{aligned} \text{Max Min} \quad & \{d_1,\ldots,d_n\} \\ \text{s.t.} \quad & p_1\frac{U_1}{d_1} + \ldots + p_n\frac{U_n}{d_n} = 1, \\ & 0 \le d_i \le D_i, i = 1,\ldots,n. \end{aligned} \tag{19}$$

Since we will increase $d_1, d_2, \ldots, d_n$ together, and the download capacity is not the bottleneck, the problem is equivalent to replacing the objective function as

$$\begin{aligned} \text{Max} \quad & d \\ \text{s.t.} \quad & \frac{p_1U_1 + \ldots + p_nU_n}{d} = 1, \end{aligned}$$

with the same inequality constraints. This clearly gives the solution $d_1 = d_2 = \ldots = d_n = d$ as the max-min allocation:

$$\text{Type } i \text{ peer:} \qquad d_i = d \equiv p_1U_1 + \ldots + p_nU_n. \tag{20}$$

From the above rate assignment strategy, one can have the following insights:

- The system allocates the same downloading rate for all types of peers. Although the "fat" peers contribute more than the "thin" peers, they are served at the same level. This service differentiation policy is *equalitarian*, but the "fat" peers may still feel unfair about the outcome. In terms of fairness, however, it is at least better than the strategy to achieve the optimal average downloading time because under that strategy the system may serve the "fat" peers even worse than the "thin" peers.
- Since we assume the downloading is not the bottleneck for all types of peers, it requires

  $$D_i \ge p_1U_1 + \ldots + p_nU_n.$$

  This simplifying assumption may not be true in every system. When we relax this assumption and we can still use the waterfilling algorithm [11] to get $d_1, \ldots, d_n$. The result is more complicated, but it leads to similar conclusions based on our preliminary study.[1]

Under the max-min fairness policy, the average downloading time is:

$$T_{mm} = \frac{1}{p_1U_1 + \ldots + p_nU_n},$$

[1]This result will be reported in a technical report or a more completed paper later on.

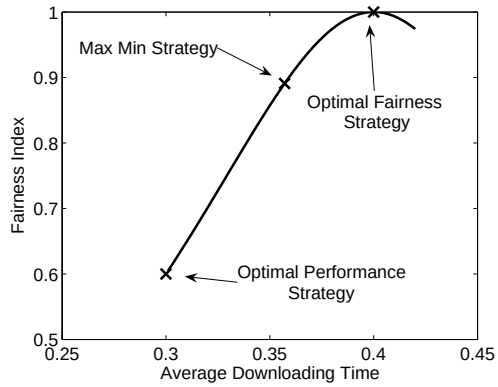

Fig. 2. Illustrating the design space in a system with two types of peers

and the Fairness index is:

$$\mathcal{F}_{mm} = \frac{(\sum_{i=1}^{n} p_i U_i)^2}{\sum_{i=1}^{n} p_i U_i^2}.$$

E. Performance and Fairness Comparison

We have calculated the performance and fairness measures of three different rate assignment strategies for BT-like file sharing systems. Their relationships are summarized in the following theorem:

Theorem *1:* In terms of the average system downloading time T, we have:

$$T_{opt} < T_{mm} < T_{fair}.$$

In terms of the fairness index, we have:

$$\mathcal{F}_{opt} < \mathcal{F}_{mm} < \mathcal{F}_{fair}.$$

Proof: Please refer to [7]. ∎

Remark: The above theorem reveals the *fundamental tussle*. Roughly speaking, the more the system differentiates the peers according to their contributions, the higher the fairness index can be achieved, but at a cost of worse performance in downloading a file. This is important for protocol designers to know since we need to realize this tradeoff and make the appropriate decision according to the objective of the application.

We use a numerical example to illustrate Theorem 1. Let us consider a BT-like file sharing system with two types of peers and the parameters are set as following: $p_1 = 0.4, p_2 = 0.6, D_1 = 5, D_2 = 6, U_1 = 4, U_2 = 2$. Both types of peers saturate the uploading capacity which means $u_1 = U_1, u_2 = U_2$, but the downloading rates are adjustable. Figure 2 illustrates the design space for all feasible downloading rate assignment strategies. In this figure, the y-axis represents the fairness measurement $\mathcal{F}$ and x-axis is the average system downloading time T. Given a feasible rate assignment strategy, we can always calculate the performance and fairness measures which correspond to a point on the curve in this figure. From this figure, one can observe that the optimal performance strategy gives the smallest average downloading time, but the lowest fairness index. Obviously the optimal fairness strategy is the best in terms of fairness, but its performance is much worse. The max-min strategy gives a tradeoff in between the former two strategies.

IV. A Family of Distributed Algorithms

In the previous sections, we present the rate assignment strategies to achieve different tradeoffs in performance and fairness. These rate assignment strategies require the knowledge of maximum uploading and downloading capacities of all peers, and assume all peers work together in an unselfish manner to achieve a particular tradeoff. In real life, peers are autonomous and have limited information. Therefore, we are interested in those distributed algorithms based on peers' local objectives, how well these algorithms achieve the desired system-wide performance metrics.

Inspired by BitTorrent and its variants, we describe a couple of generic distributed algorithms for a peer to assign its uploading and downloading rates to and from its neighbors - namely: "*selective uploading*" and "*non-discriminative uploading*". By combining these two strategies in different ratios, we are able to analyze a family of different algorithms.

Assume that a peer can potentially connect to any of the other $n-1$ peers in the system. Each peer selects a subset of other peers as neighbors to provide uploading service to them as well as obtain downloading service from them. The two neighbor selection policies are:

- *Selective uploading:* a peer will provide uploading service to n_s neighboring peers, and these neighboring peers are the top n_s peers based on their downloading rates to this particular peer. Note that this policy is implemented in the current BitTorrent protocol to encourage each peer to provide the as much uploading as possible. In BT, this is known as the *"tit-for-tat"* policy.
- *Non-discriminative uploading:* a peer will provide uploading service to n_a neighboring peers, *independent* of their downloading rates to this particular peer. This is also implemented in the BitTorrent protocol, but mostly as a way to discover neighbors who can provide the best downloading rates to the local peer. In BT, it is referred as the *"optimistic-unchoking"* policy.

We make the following assumptions about how a peer assign its rates, once neighbors are selected:

1) Each peer uses its full upload capacity to help other peers. This ensures we are considering only those solutions that are Pareto efficient. Implicitly, we are assuming that the achieved fairness level gives sufficient incentive for all peers to fully devote themselves.
2) Each peer divides its uploading capacity equally among

its neighbors. This simplifies the implementation and the analysis.

Both assumptions are based on how a BT-like protocol is implemented in practice.

In the following, we explore some important properties of these two uploading policies. In other words, we analyze the achieved downloading rates and the resulting performance and fairness tradeoff for each case.

A. Selective Uploading

Before deriving the achieved average downloading rate, we need to first address whether an equilibrium can be reached when the selective uploading policy is used by all peers. In other words, assuming each peer can determine its best neighbors instantly, will each peer's selected neighbor also select this peer as its neighbor?

Assume that in a simple, fully connected BT-like file sharing system, all peers are in the set $\mathcal{N} = \{1, 2, , \ldots, n\}$, every peer $i \in \mathcal{N}$ has uploading rate $u_i > 0$ (or $0 < u_i \leq U_i$). To finish the downloading process as quickly as possible, peers want to get the maximal downloading rate d_i. So the *utility function* for peer i is $f_i(d_i) = d_i$, where $d_i \in [0, D_i]$. According to the "tit-for-tat" policy, a peer only serves those neighboring peers from which it receives the highest n_s downloading rates, and this peer will provide upload service to each of these n_s peers with an uploading rate of u_i/n_s. Let $\mathcal{A}_i$ denote peer i's neighbor set (including i itself). This setup can be viewed as a game, where $\mathcal{A}_i$ is peer i's strategy to optimize its objective (maximum downloading rate d_i). For simplicity and without loss of generality, we sort peers according to their uploading rate u_i in an non-increasing order, so peer 1 (n) has the highest (lowest) uploading rates. Based on this ordered list, we define $\lceil \frac{n}{n_s+1} \rceil$ groups of peers as follows:

$$\begin{aligned} \mathcal{G}_0 &= \{1, 2, \ldots, n_s + 1\}, \\ \mathcal{G}_1 &= \{n_s + 2, n_s + 3, \ldots, 2(n_s + 1)\}, \\ \mathcal{G}_2 &= \{2(n_s + 1) + 1, \ldots, 3(n_s + 1)\}, \ldots . \end{aligned}$$

where $\mathcal{G}_k$ is the group of peers with the kth highest uploading capacity. The following theorem says the peers will form cliques in the equilibrium.

Theorem *2:* Using selective uploading, the selected neighbor sets (of all peers) reach an equilibrium, $\mathcal{A}_i = \mathcal{G}_k$ if $i \in \mathcal{G}_k$, and this is a Nash Equilibrium.

Proof: Please refer to [7]. ∎

Corollary *1:* When $u_1 < u_2 < \ldots < u_n$, which implies that uploading rates of all peers $i \in \mathcal{N}$ are unique, the Nash equilibrium of the game is unique.

Proof: The uniqueness results from the fact that the grouping based on $\mathcal{G}_k$'s defined above is unique. ∎

In this Nash Equilibrium, the average downloading rate of peer $i \in \mathcal{G}_k$ is:

$$d_i = \frac{1}{n_s} \sum_{j \in \mathcal{G}_k \setminus \{i\}} u_j. \tag{21}$$

A special case to consider is that if for all $i \in \mathcal{G}_k$, the difference of their uploading rates is very small (i.e., they all have similar uploading rates), then based on (21), one can claim that the average downloading rate is:

$$d_i \approx u_i. \tag{22}$$

The analysis above shows us that if all peers use the "tit-for-tat" policy as their peer selection algorithm, eventually the system will reach the Nash equilibrium, and if we maximize the uploading rates of all peers, i.e., $u_i = U_i$ for all $i \in \mathcal{N}$, this ensures the optimal fairness index $\mathcal{F} = 1$. The justification of the above claim is that this is *exactly* the rate assignment as specified in (17).

B. Non-discriminative Uploading

When the non-discriminative uploading is used, a peer will "*randomly*" choose $n_a \geq 1$ peers to provide upload service, and each of these neighboring peers will receive an uploading rate of u_i/n_a. Since there are n peers in the system, the probability that peer j is chosen by peer i is $n_a/(n-1)$. So the average total downloading rate for peer i is:

$$\begin{aligned} d_i &= \sum_{j \in \mathcal{N} \setminus \{i\}} \frac{u_j}{n_a} \cdot \frac{n_a}{n-1} \approx \frac{\sum_{j \in N} u_j}{n} \quad \text{(for large } n\text{)}, \\ &= \bar{u}. \end{aligned}$$

Here, $\bar{u}$ is the average uploading rate for all peers in the system. By using the non-discriminative uploading, all peers get the *same* downloading rate. In practice, this can be achieved by a peer *constantly* changing its neighboring set, so $\mathcal{A}_i$ is non-stationary, but the average downloading rate ($\bar{u}$) is stationary.

Again, if all peers maximize their uploading rates, i.e., $u_i = U_i$ for all $i \in \mathcal{N}$, this implies that the non-discriminative uploading policy actually achieves the max-min fairness criterion. The above claim is justified because this is *exactly* the rate assignment as indicated by (20).

From Theorem 1, we know that when the system provides the max-min fairness in the downloading rates, it has a better performance than system which provides optimal fairness (i.e., $T_{mm} \leq T_{fairness}$). But this is achieved at the cost of losing some fairness. A side-effect of the non-discriminative uploading is that it makes *free-riding* possible, or in other words, assumption 1 above may no longer hold.

C. Design Knobs

To explore the whole design space of the BT-like protocol, we consider the following design knobs: (a) uploading rate of a peer (i.e., u_i); (b) number of uploading neighbors based on the selective uploading policy (i.e., n_s), and (c) number of

uploading neighbors based on the non-discriminative uploading policy (i.e., n_a). As a matter of fact, one can express the average downloading rate of peer i, where $i \in \mathcal{N}$ as:

$$d_i = \frac{n_s}{n_s + n_a} u_i + \frac{n_a}{n_s + n_a} \bar{u}. \quad (23)$$

It is easy to check that when $n_a = 0$, this becomes the selective uploading policy. When $n_s = 0$, this becomes the non-discriminative uploading policy. We can find that n_s is used to "tune" the degree of fairness; while n_s is used to "tune" the degree of performance. By adjusting the combinations of u_i, n_s and n_a, the system can satisfy various degree of performance and fairness requirements.

It is interesting to point out that the current BitTorrent protocol is one particular implementation in the whole design space. In particular, the BitTorrent protocol has both the "tit-for-tat" policy and the "optimistic-unchoking" policy, and the official BitTorrent protocol specifies $n_s = 4$ and $n_a = 1$. So one can conclude that, the BitTorrent protocol puts more emphasis on the fairness (or incentive) measure.

V. Performance Evaluation

In this section, we quantify the merits and validate the claimed properties of BT-like file sharing protocols based on experimental evaluation. In particular, simulation studies are used to evaluate the performance of the different rate assignment strategies as well as the family of distributed algorithms in a dynamic system (i.e., peers randomly arrive and leave after the file download). For some network measurements (i.e. real-life BitTorrent file downloading sessions), please refer to our technical report [7].

A. Experiment 1: Verification

In the first experiment we show that our mathematical model presented in Section II can capture the behavior of BT-like P2P file sharing system. We simulate the BT-like system with heterogeneous (two types of) peers. We divide the file into 100 chunks, and each chunk needs one unit of time to be transferred between two peers. In the system, peers arrive according to a Poisson process with an average arrival rate $\lambda = 1.5$. Once the peer joins the system, the peer is fully connected with all other peers. Peers will leave the system immediately after they collect all 100 chunks from their neighbors. A new peer has probability of 0.4 to be a "fat" peer and 0.6 to be a "thin" peer. For "fat" peers, we have $D_1 = 5$, $U_1 = 4$, while for "thin" peers, we have $D_2 = 6$ and $U_2 = 2$. These system parameters are the same as in the example we have illustrated in Figure 2. We use different random seeds to start the simulation. For the theoretic prediction, we compute the average downloading/uploading rate $\bar{d}_i$ and $\bar{u}_i$ for each class of peers and substitute them into (10) and (12); these values are tabulated in the "Numerical" column of Table I and II. For the simulation result (the "Simulation" column of the Tables), we record all peers' actual downloading times and share ratios to get the average downloading time and fairness index of the system. Then we compare the theoretical prediction with the simulation results to see how close is the prediction on T and $\mathcal{F}$ under different values of d_i and u_i. From the tables, we observe that the percentage error is small even under the dynamic setting. One can conclude that our mathematical model can accurately predict the performance of T and $\mathcal{F}$ for a BT-like file sharing system, based on the class average downloading/uploading rates.

rate assignment				T		
d_1	d_2	u_1	u_2	Numerical	Simulation	% Err
1.91	5.39	3.43	1.81	32.1	33.9	5.4%
0.98	4.65	1.88	1.73	53.1	52.9	1.7%
2.64	3.48	3.78	1.89	32.4	35.1	8.3%

TABLE I
COMPARING NUMERICAL & SIMULATION RESULTS FOR AVERAGE DOWNLOADING TIME T

rate assignment				$\mathcal{F}$		
d_1	d_2	u_1	u_2	Numerical	Simulation	% Err
1.91	5.39	3.43	1.81	0.62	0.63	1.4%
0.98	4.65	1.88	1.73	0.63	0.58	7.5%
2.64	3.48	3.78	1.89	0.81	0.77	4.6%

TABLE II
COMPARING NUMERICAL & SIMULATION RESULTS FOR FAIRNESS INDEX $\mathcal{F}$

B. Experiment 2: Fundamental Tradeoff between T and $\mathcal{F}$

As we discussed, there is a fundamental tradeoff between performance and fairness for a BT-like file sharing system. We illustrate this tradeoff via simulation. For the system parameter, we use the same settings as in Exp 1. In this simulation, we test many different strategies under the capacity constraint same as in Exp. 1: $D_1 = 5$, $D_2 = 6$, $U_1 = 4$, $U_2 = 2$ and pick out three strategies that yielded the highest performance, highest fairness and max-min downloading rate respectively. These three strategies and their performance are summarized in the following table: One can compare the

d_1	d_2	u_1	u_2	T	$\mathcal{F}$
1.95	5.51	3.72	1.61	30.5	0.53
3.36	3.37	3.78	1.90	36.3	0.83
3.70	1.88	3.62	1.84	42.6	0.98

TABLE III
FUNDAMENTAL TRADEOFF BETWEEN T AND $\mathcal{F}$

performance and fairness tradeoff from Table III with the numerical results shown in Figure 2 (Note since the simulation used a chunk number of 100, the average downloading time

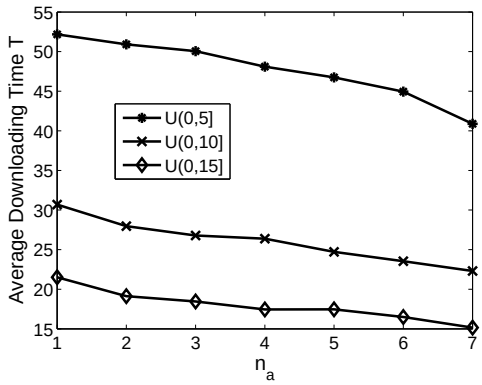

Fig. 3. n_a vs T

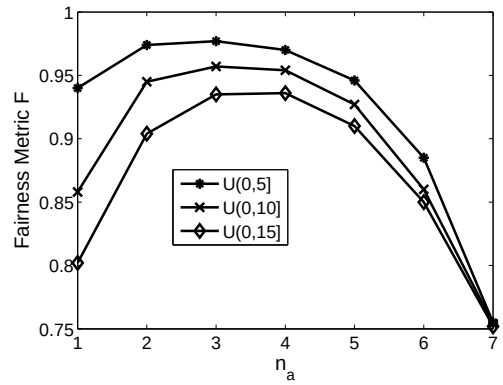

Fig. 4. n_a vs $\mathcal{F}$

in the table is equivalent to 100 times the value shown in Figure 2). The table shows that when a strategy achieves better performance (i.e., low value of T), its fairness index is also low. In other words, good performance is obtained at the cost of fairness. For strategy that achieves high fairness index, the performance on average file downloading time is also high. This is the important tradeoff that designers need to keep in mind. Note that, in our analytical derivation, we assume all peers can saturate their uplink capacity. From the results in Exp. 1 and Exp. 2, this assumption is validated since real downloading rates are quite close to the uplink capacity of both types of peers, although there is still a small gap (e.g. between the realized uploading rates and the actual capacities). From further experimentation, we observe that if one divide the file into more chunks, it improves the result since the achieved uploading rates are even closer to the uplink capacities.

C. Experiment 3: Performance Under Various Design Knob Settings.

As we have shown, the number of selective and non-discriminative uploading n_s and n_a can be considered as the design knob for BT-like protocol designer to achieve various degree of fairness and performance requirements. In this experiment, we examine the effect of the design knob under different network settings. We carry out simulation to see the performance of a *dynamic* BT-like file sharing system. In the simulation, we set the number of selective uploading neighbors as n_s. We also use n_a non-discriminative uploading neighbors to find a better peer to connect to. Then we vary the number of non-discriminative upload n_a from 1 to its maximum value but keep $n_s+n_s = 7$. For each time unit of the simulation, there are $\lambda = 4$ new peers joining the file sharing system. Each of these peers will have its own uplink capacity D_i. Here we consider several cases where the upload capacity U is uniformly distributed in the following range: (a) $(0, 5]$, (b) $(0, 10]$, (c) $(0, 15]$. Fig. 3 illustrates the average downloading time T for these different settings. The larger the capacity range is, the better performance the system achieves. The reason is that peers have better uplink resource to contribute to the system. Note that the average downloading time will improve when larger n_a are used. It means that as the system shifts from the primarily selective uploading strategy to the non-discriminative uploading strategy, the lower the average system downloading time become. This observation confirms our theoretic result that the max-min allocation strategy can perform better than the most fair strategy (assuming peers do not loose incentive to contribute).

From Fig. 4, one finds that when $n_a = 1$ the fairness metric is quite low. But our theoretic result states that the "tit-for-tat" policy can ensure high degree of fairness. So what is the justification of this phenomenon? From the viewpoint of implementation, one can justify this outcome. In this simulation, peers only have the local information and they do not connect to *all* other peers in the system. So a peer does not know which neighbor can provide better uploading service. In this case, the non-discriminative uploading is helpful so as to explore the potential good neighbors. In fact, periodically, if the n_a connection has a better downloading bandwidth than the worst of the n_s connections, this "selective" connection will be dropped, and the best of the n_a connections will be switched to the "selective" class and the peer randomly selects a new peer as a non-discriminative neighbor. When n_a is 1, it means that the peer can only use one and only one non-discriminative connection to discover other peers with high uploading rate. In this simulation, the average number of peers in the system is around 100. So only one non-discriminative connection is not sufficient for peers to find a "good neighbor". From the figure, we know that when $n_a = 2$, $\mathcal{F}$ improves dramatically, which means that two to three non-discriminative connections are sufficient to discover good neighbors in this network. From this perspective, we can understand why the fairness metric gets worse when n_a is too small. When we increase n_a, it means the system is more biased toward non-discriminative uploading, so the fairness metric will decrease (as predicted by our mathematical model). Lastly, note that using the same n_a, the system that has the largest capacity range is the worst in terms of fairness. So fat peers are actually contributing more to the system than those thin peers.

VI. **Related Work**

There are number of study on the performance of BT-like file sharing system . In [21], a coarse-grain Markovian model is studied numerically to study the service capacity of the BT-like file sharing system. In [16] a fluid model is proposed to overcome the computation problem in [21]. In [13], a detailed Markovian model is proposed to investigate the scalability and effectiveness of a P2P system. The service availability when file popularity changes is analyzed in [9]. In [8] a model of BT systems with heterogeneous peers is studied.

Service differentiation in a P2P network (though not BT-like file sharing system) are also widely studied. In [11], a game theoretic approach is proposed to provide incentive and service differentiation in P2P networks. In [22], an incentive mechanism is discussed to allocate bandwidth resource 'fairly' in a P2P system by three different service differentiation policies. In [5], the steady-state performance of multi-class BitTorrent-like systems with service differentiation is analyzed. In [14] a auction framework is presented to study the resource sharing in P2P networks. In [1], [10], different incentive mechanisms are proposed to overcome the free riding problem in BT systems. Nevertheless, all the previous work only address either the performance or the fairness (incentive) issues. To our knowledge, we are the first to focus on the fundamental tradeoff between performance and fairness in a BT-like file sharing system, and how protocols can be realized in the design space.

VII. **Conclusion**

In this paper, through a complete characterization of the different rate assignment strategies, we show the fundamental and delicate tradeoff between performance and fairness for BT-like protocols. We show that the current BT-protocol is only one particular point in the whole design space. We use the fairness index as a measure of *incentive compatibility*, which is crucial for BT-like file sharing protocol. We show the rate assignments that optimize (a) average downloading time, or (2) perfect fairness, or (c) max-min allocation. To realize these different possible tradeoffs in the design space, we propose a simple design knob which can be implemented in a distributed manner. We quantify the performance merits, both in average downloading time and fairness, as we vary the design knob. Lastly, performance evaluation is carried out to quantify the merits and properties of these BT-like protocols.

VIII. Acknowledgements

Dah Ming Chiu is supported in part by the RGC Earmark Grant $4232/04E$. John C.S. Lui is supported in part by the RGC Earmark Grant $4186/03E$. This work is affiliated with the Microsoft-CUHK Joint Laboratory for Human-centric Computing and Interface Technologies.

References

[1] E. Adar. Drawing crowds and bit welfare. *SIGecom Exch.*, 5(4):31–40, 2005.

[2] A. R. Bharambe, C. Herley, and V. N. Padmanabhan. Analyzing and improving a bittorrent networks performance mechanism. In *IEEE Infocom*, Barcelona, Spain, 2006.

[3] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[4] D.-M. Chiu, R. Yeung, J. Huang, and B. Fan. Can network coding help in p2p networks. In *Second Workshop of Network Coding, in conjunction with WiOpt*, Boston, 2006.

[5] F. Clvenot-Perronnin and K. R. P. Nain. Multiclass p2p networks: Static resource allocation for service differentiation and bandwidth diversity. In *IFIP WG7.3 Performance*, Juan-les-Pins, France, 2005.

[6] B. Cohen. Incentives build robustness in bittorrent. http://bitconjurer.org/BitTorrent/bittorrentecon.pdf, May 2003.

[7] B. Fan, D.-M. Chiu, and J. C.S.Lui. The delicate tradeoffs in bittorrent-like file sharing protocol design. Technical report, The Chinese Univeristy of Hong Kong, 2006.

[8] B. Fan, D.-M. Chiu, and J. C. Lui. Stochastic analysis and file availability enhancement for bt-like file sharing systems. In *Fourteenth IEEE International Workshop on Quality of Service (IWQoS) 2006*, 2006.

[9] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang. Measurements, analysis, and modeling of bittorrent-like systems. In *Internet Measurement Conference*, 2005.

[10] S. Jun and M. Ahamad. Incentives in bittorrent induce free riding. In *SIGCOMM 05 Workshop P2P-ECON*, 2005.

[11] T. Ma, C. Lee, J. C. Lui, and D. K. Yau. Incentive and service differentiation in p2p networks: A game theoretic approach. *to appear in IEEE/ACM Transactions on Networking*.

[12] T. B. Ma, S. C. M. Lee, J. C. S. Lui, and D. K. Y. Yau. Incentive resource distribution in p2p networks. In *The 24th International Conference on Distributed Computing Systems (ICDCS)*, March 2004.

[13] L. Massoulie and M. Vojnovic. Coupon replication systems. In *Proc. ACM SIGMETRICS*, 2005.

[14] M. Meo and F. Milan. A rational model for service rate allocation in peer-to-peer. In *IEEE Infocom*, 2005.

[15] J. Mundinger, R. Weber, and G. Weiss. Analysis of peer-to-peer file dissemination amongst users of different upload capacities. Poster at IFIP WG7.3 Performance, October 2005.

[16] D. Qiu and R. Srikant. Modeling and performance analysis of bittorrent-like peer-to-peer networks. In *Proc. ACM SIGCOMM*, 2004.

[17] B. Radunovic and J.-Y. Le Boudec. A Unified Framework for Max-Min and Min-Max Fairness with Applications. In *40th Annual Allerton Conference on Communication, Control, and Computing*, 2002.

[18] R.K.Jain, D.-M. Chiu, and W. R.Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. Technical report, DEC, 1984.

[19] S. Saroiu, P. K. Gummadi, and S. D. Gribble. A measurement study of peer-to-peer file sharing systems. In *Multimedia Computing and Networking 2002 (MMCN 02)*, January 2002.

[20] M. Shreedhar and G. Varghese. Efficient fair queueing using deficit round robin. In *SIGCOMM*, pages 231–242, 1995.

[21] X. Yang and G. de Veciana. Service capacity of peer to peer networks. In *Proceedings of IEEE INFOCOM*, 2004.

[22] X. Yang and G. de Veciana. Performance of peer-to-peer networks: Service capacity and role of resource sharing policies. *Performance Evaluation – Special Issue on Performance Modeling and Evaluation of P2P Computing Systems*, 63:175–194, March 2006.