The Design Trade-offs of BitTorrent-like File Sharing Protocols

Bin Fan[†]   John C.S. Lui[†]   Dah-Ming Chiu[‡]

Abstract—The BitTorrent (BT) file sharing protocol is very popular due to its scalability property and the built-in incentive mechanism to reduce free-riding. However, in designing such P2P file sharing protocols, there is a fundamental trade-off between keeping fairness and providing good performance. In particular, the system can either keep peers (especially those resourceful ones) in the system for as long as possible so as to help the system to achieve better performance, or allow more resourceful peers to finish their download as quickly as possible so as to achieve fairness. The current BT protocol represents only one possible implementation in this whole design space. The objective of this paper is to characterize the design space of BT-like protocols. The rationale for considering fairness in the P2P file sharing context is to use it as a measure of willingness to provide service. We show that there is a wide range of design choices, ranging from optimizing the performance of file download time, to optimizing the overall fairness measure. More importantly, we show that there is a simple and easily implementable design knob so that the system can operate at a particular point in the design space. We also discuss different algorithms, ranging from centralized to distributed, in realizing the design knob. Performance evaluations are carried out, both via simulation and network measurement, to quantify the merits and properties of the BT-like file sharing protocols.

Index Terms—BitTorrent, File Sharing Protocol, Performance, Fairness, Incentive Mechanism.

I. Introduction

In the past few years, a class of peer-to-peer (P2P) content distribution protocols, typified by a P2P file sharing application called BitTorrent (BT) [1], has been receiving a lot of attention in the computer networking research community. Unlike the conventional client/server style of content distribution in which performance usually degrades as the number of clients increases, the performance (i.e., file downloading time) of BT-like file sharing systems may actually improve when the number of participants increases. The main reason which makes BT scalable is credited to its *cooperative* mechanism. It can be explained intuitively as follows. The original file content is partitioned into many small pieces. Each peer can get the content either from the server who holds the original file, 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 in the system so as to

maximize its ability to serve others hence also the service it will receive. By coupling the service each peer can receive to its contribution to others peers, the BT protocol successfully makes each peer play the 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 accordingly.

Although the performance of BT-like systems scales well with the peer population, such kind of protocols may face the throughput-fairness dilemma which can be explained as follows. Peers that participate in a BT file sharing process are likely heterogeneous, i.e., they have different uploading/downloading capacities. A *fat* (or resourceful) node which connects to the Internet via the latest broadband access technology, typically has a high uploading capacity. On the other hand, a *thin* (or less resourceful) node is one that has a lower uploading capacity. Therefore, the overall system throughput of a BT-like file sharing protocol depends critically on how long the resourceful peers are retained in the system so as to sustain as large an aggregated system capacity as possible.

However, keeping the resourceful peers in the system longer than necessary so as to serve others seems inherently unfair. Note that the upload capacity is usually a parameter controllable by a peer. If offering a high uploading rate leads to receiving poorer 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 provides to the network (in a BT system, contribution refers to uploading). The more a protocol offers such service differentiation, the more likely to succeed.

In this paper, we formally study this fundamental trade-off 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 desire because all peers want to finish the file downloading process as soon as possible.
2) *Fairness*: To make the system scalable, peers need to contribute (via uploading to other peers) in order to receive service. A well-designed protocol should encourage peers to contribute as much as possible: those peers who contribute more should receive a better service (i.e., achieve a smaller average file downloading time) than those who contribute less.

In this paper, we first formally define the *performance* and *fairness* metrics, and use these two measures to analyze different BT-like file sharing protocols with heterogeneous peers.

[†]Department of Computer Science & Engineering, The Chinese University of Hong Kong, Shatin, N.T. Hong Kong; {bfan,cslui}@cse.cuhk.edu.hk.
[‡]Information Engineering Department, The Chinese University of Hong Kong, Shatin, N.T. Hong Kong; dmchiu@ie.cuhk.edu.hk.

In particular, we address the following important questions:

1) If a protocol optimizes performance, what would happen to the overall fairness?
2) If fairness is optimized, what happens to the overall system performance?
3) What degree of performance and fairness the current BT file sharing protocol is achieving?
4) Can we characterize the different trade-offs between performance and fairness?
5) Is there a distributed approach to implement the protocol so as to realize different operating points in the trade-offs between performance and fairness?

In this study, we assume that peers share each others' upload capacity. Peers arrive randomly in joining the system, and leave after completing the file download. For the general case of multiple types of peers (i.e., each type of peer is defined by its upload/download capacity), the complete spectrum of performance versus fairness trade-offs is analyzed.

The organization of this paper is as follows. In Section II, we present a mathematical model and define the performance measures which we will use to illustrate the whole design space. In Section III, we describe several rate assignment strategies to achieve different forms of optimality, namely, (1) optimal downloading time, (2) optimal fairness, and (3) max-min fairness. In Section IV, we present a *"distributed framework"* via uploading policy and show how this can achieve different operating points in the whole design space. In Section V, performance evaluation is carried out to quantify the merits of our proposed method. Related work is given in Section VI and finally, Section VII concludes.

II. Mathematical Model

A. A Basic Model of Uplink Sharing

Let us first describe a basic model of a BT-like file sharing system. The system is designed to disseminate a specific file among the set of peers $\mathcal{N} = \{1, 2, \ldots, n\}$. Without loss of generality, we assume the file size to be unity. A peer, say i, in this file sharing session has its upload capacity denoted as U_i and download capacity D_i. In other words, U_i is the maximum uploading rate and D_i is the maximum downloading rate that peer i can sustain. For the BT-like systems that we consider in this paper, we make the following *asymmetric up/down link assumption:*

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

which is true for most of the recent Internet access technologies and this inequality has been verified by the measurement studies in [2], [3].

To ease the dissemination, this file is divided into M distinct chunks. The uploading utilization is highly depended on the chunk scheduling efficiency of peers. In this paper we study the system under the *perfect scheduling assumption*, which indicates that each peer always has some chunks useful so that it can upload these chunks to one or more other peers. This assumption can be justified when M is very large (the

size of each chunk is very small), so it will cost little time for an empty peer to download one or more chunks. After that, this peer can serve what it owns to other peers in the system. In this way, in a short time after joining the system, a peer can provide its uploading service to the system. Let u_i denote the actual uploading rate of peer i. This uploading rate has to satisfy the upload capacity constraint:

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

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

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

From the system's perspective, the total downloading rates of all peers must be equal to the total uploading rates, or

$$\sum_{i \in \mathcal{N}} d_i = \sum_{i \in \mathcal{N}} u_i. \tag{4}$$

Note that this model is similar as the *uplink capacity sharing model* studied in [4], [5], in which the network is assumed not to be the bottleneck, but the limit to the throughput is the uplink capacities. In this paper we also make a similar capacity sharing assumption where we neglect the bottleneck or topology inside the network, but assume the limit of the system are both uplink and downlink capacity.

Since there are $n = |\mathcal{N}|$ peers in the system, any rate vectors $u = [u_1, u_2, \ldots, u_n]$ and $d = [d_1, d_2, \ldots, d_n]$ that satisfy the above constraints of (2) to (4), are considered as a *feasible rate allocation strategy*. Then the performance of a P2P file sharing system can be evaluated based on the feasible rate allocation strategy of the uplink capacity resource.

It is important to point out that the uplink capacity sharing model involves an assumption of perfect scheduling. But in reality, it is possible that the content one peer has may not be useful for other peers, so this peer cannot provide any upload service to the others. 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 basic model in the following aspects:

- In [4], [5], all peers join the system simultaneously at the beginning and depart from the system together after the last peer finishes its downloading process. In here, we allow dynamic arrival and departure of peers. Specifically, peers arrive at the system with an average rate λ. Once a peer is in the system, it performs the uploading and downloading operations at the same time. Peers will remain in the system until they finish downloading the entire file. Peers won't abort during the downloading session. In this paper, we consider the system under *no server assumption*, which means that upon completion of a file download, a peer will leave the system immediately

since there is no incentive for a peer to become a seeder after downloading the file. This assumption is compatible to the existing measurement-based study of current BT protocol such that most of the peers leave the system in a very short time after downloading the entire file [6].

- 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 that 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 the type-i peers 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). Due to the previous asymmetric up/down link assumption, we have $U_i < D_i$, for $i = 1, \ldots, n$. Besides, we have the following downlink capacity ordering assumption that for these different types of peers, they have the following relationship in their downloading and uploading capacities:

$$U_1 > U_2 > \ldots > U_n; \quad D_1 > D_2 > \ldots > D_n, \quad (5)$$

this assumption is reasonable given most of currently deployed technologies. As we will show, based on this assumption, we can characterize the relationship between different resource allocation policies in later section. 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 the physical constraints in Equations (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 of presentation, in the remaining of this paper we will refer to type-1 peers as *"fat"* peers, and all the other types of peers as *"thin"* peers.

Fig. 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 a higher upload capacity and type-2 ("thin") peers which have a lower upload capacity.
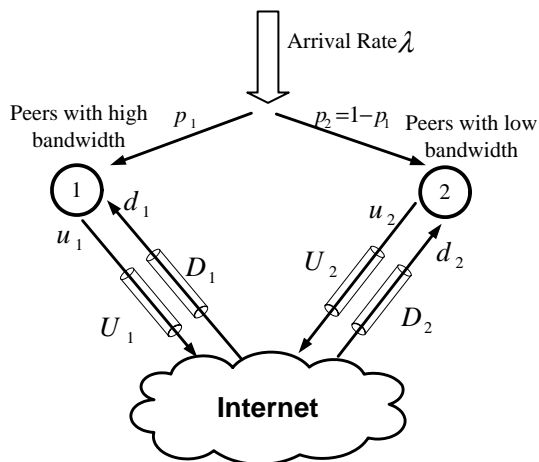

Fig. 1. Illustration of the Dynamic Model with two types of peers.

C. Performance Measure

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

$$\mathcal{C}(t) = \sum_{i=1}^{n} u_i(t) N_i(t). \quad (6)$$

Here $u_i(t) N_i(t)$ is the uploading capacity brought by type-i peers. From Eq. (6), one can find that the service capacity is also related to the number of different types of peers and the distribution of peers' uploading capacity.

Since a file sharing process usually takes a significant amount of time, 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 (N_i) and their average file downloading time (T_i) as:

$$N_i = (p_i \lambda) \cdot T_i = \frac{p_i \lambda}{d_i} \quad \text{for } i \in \{1, \ldots, n\}. \quad (7)$$

Here, T_i and d_i are the average downloading time and downloading rate for type-i peer. Substituting Eq. (7) into (6) for $i = \{1, \ldots, n\}$, the steady state system service capacity $\mathcal{C}$ becomes:

$$\mathcal{C} = p_1 \lambda \frac{u_1}{d_1} + \ldots + p_n \lambda \frac{u_n}{d_n}. \quad (8)$$

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 normalized to one), and we get the following equation:

$$p_1 \frac{u_1}{d_1} + \ldots + p_n \frac{u_n}{d_n} = 1. \quad (9)$$

Let us define $c_i = u_i/d_i$. Using the BT terminology, c_i is called *"share ratio"* of type-i peers, and it 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 more contribution to the system. Using the notation of c_i, Eq. (9) becomes:

$$p_1 c_1 + \ldots + p_n c_n = 1, \quad (10)$$

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

Now, applying the Little's result, the average system downloading time T for all peers can be expressed as:

$$T = \frac{N_1 + \ldots + N_n}{\lambda} = \frac{p_1}{d_1} + \ldots + \frac{p_n}{d_n}. \quad (11)$$

The above two equations provide 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 different types of peers, namely, $\{u_i, d_i | i = 1, \ldots n\}$.

So far, we have derived the expression for a particular metric of interest, namely, average system downloading time T. In the following, let us focus on the other system measure of interest, namely, fairness.

D. Fairness Measure

The issue of *fairness* has long been studied in the networking community. For example, in [7], [8], one can find various expressions to quantify the fairness among competing entities. In here, 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 P2P system relative to the obtained service. Share ratios are more important for BitTorrent protocol than other peer-to-peer file sharing systems, because some BT systems require peers to maintain a minimum share ratio. Due to the physical meaning of share ratio, a well-designed system should try to make all its peers maintain the same share ratio.

To quantify the fairness of a system according to the peers' share ratios, we have the following Fairness Index:

$$\mathcal{F}(x_1, \ldots, x_n) = \frac{(\sum_{i=1}^{n} x_i)^2}{n(\sum_{i=1}^{n} x_i^2)}. \tag{12}$$

This index measures how equal the assignment $(x_1, \ldots, x_n)$ is. This fairness measure has the following attractive properties:

- $\mathcal{F}$ is *independent of scale*. If we have $x_i' = a \cdot x_i$ for $i = 1, \ldots, n$ and $a > 0$, then $\mathcal{F}(x_1, \ldots, x_n) = \mathcal{F}(x_1', \ldots, x_n')$. In other words, scaling the allocated resource will not affect the measure $\mathcal{F}$.
- $\mathcal{F}$ is *continuous*. If we change any x_i for $i \in \{1, \ldots, n\}$, $\mathcal{F}$ will be changed continuously.
- $\mathcal{F}$ is *independent of population size*. It can be applied to any number of peers, as long as $n \geq 1$.
- $\mathcal{F}$ is a *bounded* function between 0 and 1. This facilitates us to precisely measure the degree of fairness. The greater value $\mathcal{F}$ is, the more fair the assignment is. $\mathcal{F}$ equals to 1, if and only if $x_1 = \ldots = x_n$. This indicates the perfect fairness of the assignment because all measured elements are the same.

The above properties of $\mathcal{F}$ are useful for us to quantify the fairness of share ratios among all peers. Assume that the system reaches the steady state, all together K peers have finished downloading, and K_i is the number of type-i peers which finished downloading. Therefore, $K_i = p_i \cdot K$. Substitute $x_i = c_i$ and use Eq. (10), we have:

$$\mathcal{F} = \frac{(K_1 c_1 + \ldots + K_n c_n)^2}{K \cdot (K_1 c_1^2 + \ldots + K_n c_n^2)} = \frac{(p_1 c_1 + \ldots + p_n c_n)^2}{p_1 c_1^2 + \ldots + p_n c_n^2}$$
$$= \frac{1}{p_1 c_1^2 + \ldots + p_n c_n^2}. \tag{13}$$

It is important to point out that from the properties of the Fairness Index, we know that when $c_1 = \ldots = c_n$, $\mathcal{F}$ reaches its maximum value. In other words, when all peers have the same share ratio, the system reaches its maximum fairness measure.

So far we have defined two metrics for the P2P uplink sharing problem, performance according to the average delay T and the fairness measure $\mathcal{F}$, and both expressed in terms of 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 rate assignments.

III. Rate Assignment Strategies

In this section, we present several rate assignment strategies to realize different operating points in the design space of the BT-like file sharing protocol.

A. Uploading Rate

Since peers are 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 trade-off can be achieved.

First, let us investigate the uplink rate assignment for all peers. BitTorrent protocol is generally considered very effective in file content distribution. Both experimental results [9] and measurement results [10] indicate that BitTorrent performs near-optimally in terms of uplink bandwidth utilization. To achieve good performance, we have

$$u_i = U_i \qquad \text{for } i = 1, \ldots, n, \tag{14}$$

which implies that due to the efficiency of BT protocol, the upload capacity of type-i peers is always saturated.

Let us now derive the feasible settings of d_i when the system wants to (a) minimize the average downloading time T, or (b) maximize the fairness measure $\mathcal{F}$, or (c) achieve the "*max-min*" allocation of the downloading rate.

B. Rate Assignment for the 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 Eq. (11), one needs to solve the following constrained optimization problem:

$$\begin{aligned} \min \quad & T = \frac{p_1}{d_1} + \ldots + \frac{p_n}{d_n} \\ \text{s.t.} \quad & p_1 \frac{U_1}{d_1} + \ldots + p_n \frac{U_n}{d_n} = 1, \\ & 0 \leq d_i \leq D_i, \qquad i = 1, \ldots, n. \end{aligned} \tag{15}$$

The first equality constraint is the steady state condition given by Eq. (9). The other inequality constraints are due to its

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

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

The Karush-Kuhn-Tucker (KKT) conditions [11] for the above optimization are:

$$\begin{cases} \frac{\partial L}{\partial d_i} = 0; \\ 0 \le d_i \le D_i, \quad i = 1, \ldots, n; \\ p_1 \frac{U_1}{d_1} + \ldots + p_n \frac{U_n}{d_n} = 1; \\ \lambda_i \ge 0, \quad i = 1, \ldots, n. \end{cases}$$

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

$$\begin{aligned} \text{Type-1 peer}: \quad & d_1 = \frac{p_1 U_1}{1 - \sum_{i=2}^{n} p_i \frac{U_i}{D_i}}, \\ \text{Type-}i \text{ peer}: \quad & d_i = D_i, \quad i = 2, \ldots, n. \end{aligned} \quad (16)$$

The rate assignment strategy to achieve the optimal downloading time by (16) 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 the reason why the system can achieve the optimal performance of file downloading time.

- Consider the service differentiation of uploading resource for both "fat" and "thin" peers, we have:

$$d_1 - d_i = \frac{p_1}{1 - \sum_{j=2}^{n} p_j \frac{U_j}{D_j}} U_1 - D_i. \quad (17)$$

  From the above equation, one can find that when the parameters satisfy $p_1 \frac{U_1}{D_i} + \sum_{j=2}^{n} p_j \frac{U_j}{D_j} < 1$, we have $d_1 < d_i$, which means that the "thin" peers receive *better* service than those "fat" peers eventhough "fat" peers are contributing more to the system (i.e. $U_1 > U_i$)! This indicates the potential *unfairness*, or the *cost* of achieving the optimal file downloading time.

For this policy, one can use Eq. (11) to express the achieved average file downloading time T_{opt}, which is:

$$T_{opt} = \frac{1}{U_1} + \sum_{i=2}^{n} \frac{p_i}{D_i} \frac{U_1 - U_i}{U_1}. \quad (18)$$

One can use Eq. (13) to express the fairness measure $\mathcal{F}_{opt}$ as:

$$\mathcal{F}_{opt} = \left[p_1 \left(\frac{1 - \sum_{i=2}^{n} p_i U_i / D_i}{p_1} \right)^2 + \sum_{i=2}^{n} p_i \left(\frac{U_i}{D_i} \right)^2 \right]^{-1}. \quad (19)$$

C. Rate Assignment for Optimal Fairness

Now let us consider 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:

$$c_1 = \ldots = c_n = \gamma,$$

where γ is a positive real number. Due to the steady state condition of (10), 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 a perfect fairness of $\mathcal{F} = 1$, the system needs to ensure that each type of peers can only receive as much resource as they contribute.

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

$$\text{Type-}i \text{ peer}: \quad d_i = u_i = U_i, \quad i = 1, \ldots, n. \quad (20)$$

The rate assignment strategy given in Eq. (20) means that to achieve the maximum fairness, the system needs to set the downloading rate of each peer to its uploading rate. One can observe that this service differentiation policy is fair because:

- $U_i < D_i$, therefore no peer will saturate its downlink capacity.
- The share ratio is 1 for all peers, therefore a "fat" peer can download faster than a "thin" peer since the "fat" peer contributes more. So peers have incentive to contribute more to the P2P network.

In summary, one can use Eq. (11) to derive the average downloading time for this policy as:

$$T_{fair} = \frac{p_1}{U_1} + \ldots + \frac{p_n}{U_n}, \quad (21)$$

and one can use Eq. (13) to derive the fairness metric:

$$\mathcal{F}_{fair} = 1. \quad (22)$$

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 to consider a rate assignment strategy which can achieve the *max-min* allocation index. Let us first state the definition of max-min allocation [12].

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

$$(\forall \vec{y} \in \mathcal{P}) \, (\exists s \in \{1, \ldots, n\}) \, y_s > x_s \implies$$
$$(\exists t \in \{1, \ldots, n\}) \, y_t < x_t \le x_s. \quad (23)$$

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

The max-min allocation can be constructed using a waterfilling algorithm [13], increasing the downloading rates of all types of peers d_1, d_2, ..., d_n simultaneously from zero. Since $D_1 > D_2 > \ldots > D_n$, it is possible that during this waterfilling procedure, some types of peers may saturate their downloading capacity (i.e., $d_i = D_i$). Then we do not increase the downloading rates of these saturated peers any more. We use the following pseudo code to describe the strategy:

Pseudo Code of Waterfilling

```
1.   d₁ = ... = dₙ = 0
2.   S ← {1, 2, ..., n}
3.   repeat{
4.     for i in S
         /* δ is waterfilling rate*/
5.       if dᵢ + δ ≥ Dᵢ {
6.         dᵢ ← Dᵢ
7.         S ← S \{i}
       }
8.       else { dᵢ ← dᵢ + δ }
10.  } until (p₁ U₁/d₁ + ... pₙ Uₙ/dₙ ≤ 1)
11.  k ← max{i|i ∈ S}
```

We increase the downloading rate of every type of peers at each rate until some peers achieve the download capacity. Since $D_1 > D_2 > \ldots > D_n$, therefore by using the waterfilling algorithm above, we can find k such that, for type-$(k+1)$, ..., type-n peers, due to their lower downloading capacities, the download link of these peers are saturated (i.e. $d_i = D_i, i \in \{k+1, \ldots, n\}$). But for the other types of peers (from type-1 to type-k), they achieve the same downloading rate. Thus, it guarantees that the downloading rate assignment solution is a max-min allocation. Formally, the resulting downloading rate assignment will be:

$$
d_i = \begin{cases} \dfrac{\sum_{j=1}^{k} p_j U_j}{1 - \left(\sum_{j=k+1}^{n} p_j \frac{U_j}{D_j}\right)}, & i = 1, \ldots, k, \\ D_i & i = k+1, \ldots, n. \end{cases} \quad (24)
$$

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

- The system allocates type-$(k+1)$ to type-n peers their maximal downloading rate. Thus for these peers, this strategy is similar to the rate assignment which provides the optimal downloading time. The difference is for type-1 to type-k peers, they share the same downloading rate. Given $U_1 > \ldots > U_n$ and $D_1 > \ldots > D_n$, by this max-min allocation, if a type-i peer provides more uploading than type-j, this type-i peer will receive the downloading service which is no less than that of type-j peer. In terms of fairness, this strategy is at least better than the strategy of achieving the optimal average downloading time because under that strategy, the "fat" peers may have worse performance than the "thin" peers.

- Type-1 peers contribute more than the peers of type-2 to type-k, but they received service at the same downloading rate. Although all peers receive the same service, the "fat" (or resourceful) peers may feel unfair about the outcome compared to the strategy which achieves optimal fairness.

For the max-min fairness policy, one can use Eq. (11) to obtain the average downloading time as:

$$
T_{mm} = \sum_{i=1}^{k} \frac{p_i}{w} + \sum_{i=k+1}^{n} \frac{p_i}{D_i}, \quad (25)
$$

where w is defined by

$$
w = \frac{\sum_{j=1}^{k} p_j U_j}{1 - \sum_{j=k+1}^{n} p_j \frac{U_j}{D_j}}.
$$

Also we use Eq. (13) to obtain the fairness index as:

$$
\mathcal{F}_{mm} = \frac{1}{\sum_{i=1}^{k} p_i (\frac{U_i}{w})^2 + \sum_{i=k+1}^{n} p_i (\frac{U_i}{D_i})^2}. \quad (26)
$$

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 generalized fairness index, we have:

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

Proof: Proof is given in the Appendix. ∎

Remark: The above theorem reveals a *fundamental tussle:* 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 trade-off and make the appropriate decision according to the objective of the application.

To illustrate Theorem 1, let us consider a numerical example. We consider a BT-like file sharing system with two types of peers and parameters are set as the following: $p_1 = 0.4, p_2 = 0.6, D_1 = 8, D_2 = 6, U_1 = 4, U_2 = 2$. Both types of peers saturate the uploading capacity which means $u_1 = U_1, u_2 = U_2$, but the downloading rates are adjustable. Fig. 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 trade-off in between the former two strategies.
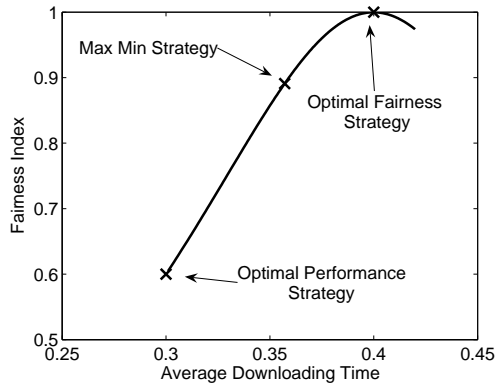

Fig. 2. Illustrating the design space in a system with two types of peers

IV. **A Family of Distributed Algorithms**

So far, we present the rate assignment strategies to achieve different trade-offs in performance and fairness. Note that these rate assignment strategies require the knowledge of maximum uploading and downloading capacities, U_i and D_i, of all peers, and assume all peers work together in an unselfish manner to achieve a particular trade-off. In real life, peers are autonomous and it is difficult to have global information of U_i and D_i. Therefore, we are interested in those distributed algorithms which are based on peers' local objectives, and to study how well these algorithms achieve the desired system-wide performance measures.

Inspired by BitTorrent system 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 distributed 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 uploading rates to this particular peer. Note that this policy is implemented in the current BitTorrent protocol to encourage each peer to provide as much uploading as possible. This is also known as the *'tit-for-tat'* policy in the BT protocol.
- *Non-discriminative uploading:* a peer will provide uploading service to n_a neighboring peers, *independent* of their uploading rates to this particular peer. This is also

implemented in the BitTorrent protocol, but mostly as a way to discover neighbors which can provide the best downloading rates to the local peer. This is referred to as the *optimistic-unchoking* policy in the BT protocol.

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 that we only consider those solutions that are Pareto efficient. Implicitly, we are assuming that the achieved fairness level gives sufficient incentive for all peers to contribute.
2) Each peer divides its uploading capacity equally among its neighbors. This simplifies the implementation and the analysis.

Both assumptions are based on how the 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 trade-off 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, if each peer has all the information, will all peers select certain peers as their neighbors to upload and make no change for their choice?

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 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 uploading set (i.e., nodes in $\mathcal{A}_i$ are those nodes that peer i will provide upload service). This setup can be viewed as a game, where peer i needs to find $\mathcal{A}_i$ which can provide peer i the best benefit (i.e., maximum downloading rate d_i). For simplicity and without loss of generality, we sort peers according to their uploading rate u_i in a non-increasing order, so peer 1 has the highest uploading rate while peer n has the lowest uploading rate. 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 states that peers will form cliques in the equilibrium.

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

Proof: If any peer $i \in \mathcal{G}_k$ decides to upload to a peer $j \in \mathcal{G}_{k'}$ where $k' < k$, peer j will not upload to peer i because all other n_s peers in group $\mathcal{G}_{k'}$ have larger or equal uploading rates than the peer i. On the other hand, if peer i provides upload to peer $j \in \mathcal{G}_{k'}$ where $k' > k$, we have the following cases: (a) When peer i has the same uploading rate u_i as peers in $\mathcal{G}_{k'} \setminus \{j\}$, peer j has no reason to change its uploading set by providing service to peer i. (b) When peer i has an uploading rate of u_i which is greater than at least one peer in $\mathcal{G}_{k'} \setminus \{j\}$, peer j will provide upload service to peer i. But the downloading rate d_i of peer i will not increase. So peer i cannot increase its utility $f_i(d_i)$ by uploading to peers in other groups. ∎

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: When u_i is unique, the grouping of $\mathcal{G}_k$ is unique. According to Theorem 2, one can easily see the uniqueness of the Nash Equilibrium. ∎

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

$$d_i = \frac{1}{n_s} \sum_{j \in \mathcal{G}_k \setminus \{i\}} u_j. \tag{27}$$

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 Eq. (27), one can claim that the average downloading rate is:

$$d_i \approx u_i. \tag{28}$$

The analysis above shows 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 Eq. (20). Note that in [10], measurement based study shows the similar "grouping effect" that peers with higher uploading rates tend to serve each other more. And this is reflected by the above theorem and corollary.

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 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 $n_a/(n-1)$. If peer i doesn't saturate its downlink capacity, the average total downloading rate for peer i is:

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

Here, $\bar{u}$ is the average uploading rate for all peers in the system. If the downlink capacities for all peers are large enough, i.e., no peer reaches its maximal downloading rate (which is often the case in practice), then 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 uploading 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 Eq. (24) when $k = n$.

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 achieved performance is at the cost of losing some fairness. Note that a side-effect of the non-discriminative uploading is that it makes *free-riding* possible.

C. Design Knobs

To explore the whole design space of the BT-like protocol, we consider the following design knobs:

- uploading rate of a peer (i.e., u_i);
- number of uploading peers based on the selective uploading policy (i.e., n_s), and
- number of uploading peers 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:

$$d_i = \frac{n_s}{n_s + n_a} u_i + \frac{n_a}{n_s + n_a} \bar{u}. \tag{29}$$

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. One can view 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 note 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, some BitTorrent implementations specify $n_s = 4$ 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 both simulation evaluation and network measurements. In particular, both simulation and network measurement (of real-life BitTorrent file downloading sessions) 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 arrive randomly and leave after the file download immediately). We develop a discrete event simulator to simulate the performance of BT-like file sharing systems.

A. Simulation 1: Verification

In the first simulation, 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 it takes one unit of time to transfer a chunk between two peers. Note that in the model, we do not consider the chunk selection but in the simulation, the data transfer takes place at the chunk level and we use the rarest-first chunk selection algorithm in BT. In our simulation, 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 $D_1 = 8$, $U_1 = 4$, 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 Fig. 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 (11) and (13); 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

B. Simulation 2: Trade-off between average downloading time T and fairness measure $\mathcal{F}$

As we discussed, there is a fundamental trade-off between performance and fairness for a BT-like file sharing system. We illustrate this trade-off via simulation. For the system

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
$\mathcal{F}$

parameter, we use the same settings as in Sim. 1. In this simulation, we test many different strategies under the capacity constraint as in Sim. 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 performance and

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 TRADE-OFF BETWEEN T AND $\mathcal{F}$

fairness trade-off from Table III with the numerical results shown in Fig. 2 (Note since the simulation used a chunk number of 100, the average downloading time in the table is equivalent to 100 times the value shown in Fig. 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 trade-off 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 Sim. 1 and 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). We have carried more simulation and we observe that if one divides the file into more chunks, it improves the result since the achieved uploading rates are even closer to the uplink capacities.

C. Simulation 3: Performance at the Nash Equilibrium

3.1: In this simulation, we want to verify the claimed property of the selective uploading policy. In particular, when the system reaches the Nash Equilibrium, peers will form cliques (based on Theorem 2) and at the equilibrium, the average downloading rate of a peer is given by Eq. (29). The question is that in a dynamic system (i.e., with stochastic arrivals and departures), whether this relationship still holds. We carry out a simulation wherein peers use the "tit-for-tat" policy in the BT-like file sharing system. In the simulation, we set the number of selective uploading neighbors as $n_s = 6$. We also use two non-discriminative uploading ($n_a = 2$) neighbors

to find a better peer to connect to. 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, uniformly distributed between $(0, 10]$. Fig. 3 depicts the relationship between the average uploading rates and the downloading rates of different peers in the system. The solid line is the prediction based on Eq. (29) with $n_s = 6$ and $n_a = 2$, while the dots are the results from the simulation. One can conclude that under the "tit-for-tat" policy, the relationship of d_i and u_i given by Eq. (29) is correct even for a dynamic arrival/departure system.
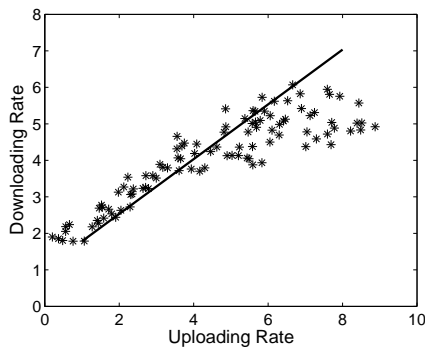

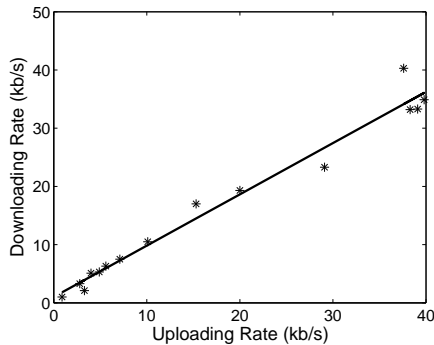Fig. 3. Downloading Rate vs Uploading Rate by Simulation

Fig. 4. Downloading Rate vs Uploading Rate by Measurement

3.2: We also carry out network measurements in the Internet to verify the claimed equilibrium. In particular, we use a BT client (version 4.1.6.) to download the file via a public torrent from a vantage point of a single peer using different uploading rate limits. We vary the uploading rate and record the corresponding downloading rates. The relationship is plotted in Fig. 4. The dots are the measurement results from the Internet. The solid line is the linear regression from all points. Official BT client version 4.1.6 sets $n_s = 3$ and $n_a = 1$ but n_s may be higher when the uploading rate is higher. One can observe that indeed $d_i \approx \alpha u_i$ holds and α is greater than $3/4$. This measurement study illustrates that the current BT-protocol is actually trying to achieve a high level of fairness (hence strong incentive for peers to contribute).

D. Simulation 4: 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 Simulation, 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. The simulation setup is similar to Exp. 3.1 but we keep $n_s + n_s = 7$. We vary the number of non-discriminative upload n_a from 1 to its maximum value. We also consider several cases where the upload capacity U is uniformly distributed in the following range: (a) $(0, 5]$, (b) $(0, 10]$, (c) $(0, 15]$.
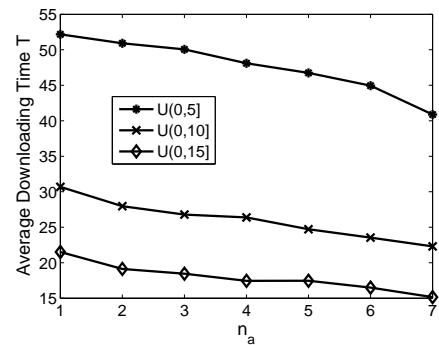

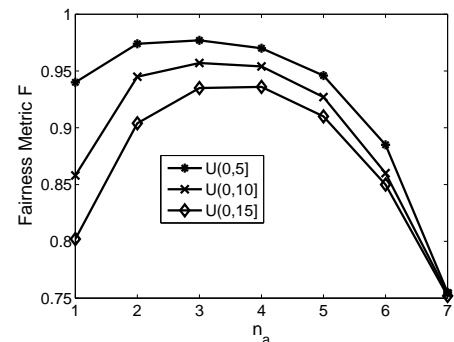Fig. 5. Average file downloading time T vs. n_a

Fig. 6. Fairness measure $\mathcal{F}$ vs. n_a

Fig. 5 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 becomes. 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. 6, 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 numerous measurement studies on the performance and behavior of actual BT system. In [14]–[16], the characteristics of BitTorrent system and users such as download performance, availability, integrity, flash crowd effect and the session features are presented by actual BitTorrent traffic. In [17], fluctuation of peer's performance in small torrents is reported and an inter-torrent collaboration is proposed to provide extra incentives for leechers. In [18], the existing peer selection and piece selection algorithms of BitTorrent, namely, *choking* and *local rarest first* are evaluated. In [19] great variability of downloading time by using BT system is reported. The authors claim that instead of network bandwidth, *"close neighbor set"* (i.e., those peers in a stable data-exchange relationship) is the major contributing factor for the variability in file downloading time. In [10], experimental investigation shows the clustering effect of similar-bandwidth peers.

There are number of analytical studies on the BT protocol' performance. In [20], a coarse-grain Markovian model is presented and authors study the service capacity of the BT-like file sharing system. In [21], a fluid model is proposed to overcome the computation problem in [20]. The Nash equilibrium as the outcome of the "tit-for-tat" policy is also discussed and the clustering effect is proved. But the relationship between fairness and performance and its design choice are not studied.

In [22], [23], a detailed Markovian models are proposed to investigate the scalability and effectiveness of a file swarming P2P system . The service availability when file popularity changes is analyzed in [17]. In [24] a model of BT systems with heterogeneous peers is studied. In [25], [26], authors present the analytical modeling of BT-like streaming systems.

Service differentiation in a P2P network (though not BT-like file sharing system) are also widely studied. In [13], a game theoretic approach is proposed to provide incentive and service differentiation in P2P networks. In [27], an incentive mechanism is discussed to allocate bandwidth resource 'fairly' in a P2P system by three different service differentiation policies. In [28], the steady-state performance of multi-class BitTorrent-like systems with service differentiation is analyzed. In [29] a auction framework is presented to study the resource sharing in P2P networks. In [30], [31], different incentive mechanisms are proposed to overcome the free riding problem.

Previous work only address either the performance or the fairness (incentive) issues. In [32], a strategic BitTorrent client is proposed to improve the performance of the peers that contribute more to the system, meanwhile achieve better fairness degree. This work is from system design and implementation perspective. To our knowledge, we are the first to propose a mathematical framework to investigate the fundamental trade-off between performance and fairness in a BT-like file sharing system, as well as how to realize a BT-like protocol in different point of the design space.

VII. **Conclusion**

In this paper, through a complete characterization of the different rate assignment strategies, we show the fundamental trade-off 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 the amount of received service when one contributes, and fairness is crucial for BT-like file sharing protocols. 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 trade-offs 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 and network measurement are carried out to validate our model and quantify the merits and properties of these BT-like protocols.

APPENDIX A
Proof of Theorem 1

From Eq. (24), we know that, under the max-min allocation, those peers which have not saturated their downloading capacities (i.e., type-1,..., type-k), they have the same downloading rate. Let us denote this downloading rate as w where:

$$w = \frac{p_1 U_1 + p_2 U_2 + \ldots + p_k U_k}{1 - (p_{k+1} \frac{U_{k+1}}{D_{k+1}} + \ldots + p_n \frac{U_n}{D_n})}.$$

First, we prove the relationship of the performance in average downloading time:

$$
\begin{aligned}
&T_{mm} - T_{opt} \\
&= \left(\sum_{i=1}^{k} \frac{p_i}{w} + \sum_{i=k+1}^{n} \frac{p_i}{D_i} \right) - \left(\frac{1}{U_1} + \sum_{i=2}^{n} \frac{p_i}{D_i} \frac{U_1 - U_i}{U_1} \right) \\
&= \sum_{i=1}^{k} \frac{p_i}{w} - \frac{1}{U_1} \left(1 - \sum_{i=k+1}^{n} p_i \frac{U_i}{D_i} \right) - \sum_{i=2}^{k} \frac{p_i}{D_i} \left(1 - \frac{U_i}{U_1} \right) \\
&= \sum_{i=2}^{k} \frac{p_i}{w} + \frac{p_1}{w} - \frac{\sum_{i=1}^{k} p_i U_i}{w \cdot U_1} - \sum_{i=2}^{k} \frac{p_i}{D_i} \left(1 - \frac{U_i}{U_1} \right) \\
&= \sum_{i=2}^{k} \frac{p_i}{w} - \sum_{i=2}^{k} \frac{p_i U_i}{w \cdot U_1} - \sum_{i=2}^{k} \frac{p_i}{D_i} \left(1 - \frac{U_i}{U_1} \right) \\
&= \sum_{i=2}^{k} p_i \left(1 - \frac{U_i}{U_1} \right) \left(\frac{1}{w} - \frac{1}{D_i} \right)
\end{aligned}
$$

Since for $i = 2, \ldots, k$, $U_i < U_1$ and $D_i > w$, we have $T_{mm} - T_{opt} > 0$. While

$$
\begin{aligned}
&T_{fair} - T_{mm} \\
&= \sum_{i=1}^{n} \frac{p_i}{U_i} - \left(\sum_{i=1}^{k} \frac{p_i}{w} + \sum_{i=k+1}^{n} \frac{p_i}{D_i} \right) \\
&= \frac{1}{w} \frac{\sum_{i=1}^{k} \sum_{j=1}^{k} p_i p_j \frac{U_j}{U_i}}{\left(1 - \sum_{i=k+1}^{n} p_i \frac{U_i}{D_i} \right)} - \frac{\sum_{i=1}^{k} p_i}{w} + \sum_{i=k+1}^{n} p_i \left(\frac{1}{U_i} - \frac{1}{D_i} \right) \\
&> \frac{1}{w} \frac{\left(\sum_{i=1}^{k} p_i \right)^2}{\left(1 - \sum_{i=k+1}^{n} p_i \frac{U_i}{D_i} \right)} - \frac{\sum_{i=1}^{k} p_i}{w} + \sum_{i=k+1}^{n} p_i \left(\frac{1}{U_i} - \frac{1}{D_i} \right) \\
&= -\frac{\sum_{i=1}^{k} p_i}{\sum_{i=1}^{k} p_i U_i} \sum_{i=k+1}^{n} p_i \left(1 - \frac{U_i}{D_i} \right) + \sum_{i=k+1}^{n} \frac{p_i}{U_i} \left(1 - \frac{U_i}{D_i} \right) \\
&= \sum_{i=k+1}^{n} p_i \left(1 - \frac{U_i}{D_i} \right) \left(\frac{1}{U_i} - \frac{\sum_{j=1}^{k} p_j}{\sum_{j=1}^{k} p_j U_j} \right) \\
&> \sum_{i=k+1}^{n} p_i \left(1 - \frac{U_i}{D_i} \right) \left(\frac{1}{U_i} - \frac{1}{U_k} \right)
\end{aligned}
$$

Since for $i = k+1, \ldots, n$, $U_i < D_i$ and $U_i < U_k$, we know $T_{mm} > T_{opt}$. In summary, we have $T_{fair} > T_{mm} > T_{opt}$.

Now let us consider the relationship of fairness:

$$
\begin{aligned}
\mathcal{F}_{mm} - \mathcal{F}_{opt} &= \frac{1}{\sum_{i=1}^{k} p_i (\frac{U_i}{w})^2 + \sum_{i=k+1}^{n} p_i (\frac{U_i}{D_i})^2} \\
&\quad - \frac{1}{p_1 (\frac{1 - \sum_{i=2}^{n} p_i U_i / D_i}{p_1})^2 + \sum_{i=2}^{n} p_i (\frac{U_i}{D_i})^2}.
\end{aligned}
$$

Since

$$
\begin{aligned}
&\left[\sum_{i=1}^{k} p_i \left(\frac{U_i}{w} \right)^2 + \sum_{i=k+1}^{n} p_i \left(\frac{U_i}{D_i} \right)^2 \right] \\
&\quad - \left[p_1 \left(\frac{1 - \sum_{i=2}^{n} p_i \frac{U_i}{D_i}}{p_1} \right)^2 + \sum_{i=2}^{n} p_i \left(\frac{U_i}{D_i} \right)^2 \right] \\
&= \left(\frac{U_1}{w} + \frac{1 - \sum_{i=2}^{n} p_i \frac{U_i}{D_i}}{p_1} \right) \left(p_1 \frac{U_1}{w} - p_1 \frac{1 - \sum_{i=2}^{n} p_i \frac{U_i}{D_i}}{p_1} \right)
\end{aligned}
$$

$$
\begin{aligned}
&\quad + \sum_{i=2}^{k} \left(\frac{U_i}{w} + \frac{U_i}{D_i} \right) \left(p_i \frac{U_i}{w} - p_i \frac{U_i}{D_i} \right) \\
&< \left(\frac{U_1}{w} + \frac{1 - \sum_{i=2}^{n} p_i \frac{U_i}{D_i}}{p_1} \right) \left(p_1 \frac{U_1}{w} - p_1 \frac{1 - \sum_{i=2}^{n} p_i \frac{U_i}{D_i}}{p_1} \right) \\
&\quad + \sum_{i=2}^{k} \left(\frac{U_1}{w} + \frac{1 - \sum_{i=2}^{n} p_i \frac{U_i}{D_i}}{p_1} \right) \left(p_i \frac{U_i}{w} - p_i \frac{U_i}{D_i} \right) \\
&= 0,
\end{aligned}
$$

thus, we have $\mathcal{F}_{mm} > \mathcal{F}_{opt}$.

Under the max-min strategy, for $i, j \in \{1, \ldots, k\}$ and $i \neq j$, $c_i = \frac{U_i}{w} \neq c_j$, so we have $\mathcal{F}_{mm} < 1 = \mathcal{F}_{fair}$. In summary, we have $\mathcal{F}_{fair} > \mathcal{F}_{mm} > \mathcal{F}_{opt}$. ∎

REFERENCES

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

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

[3] A. Bellissimo, P. Shenoy, and B. N. Levine, "Exploring the use of bittorrent as the basis for a large trace repository," UMass, Tech. Rep., June 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] 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, Juan-les-Pins, France, October 2005.

[6] J. A. Pouwelse, P. Garbacki, D. H. J. Epema, and H. J. Sips, "The bittorrent p2p file-sharing system: Measurements and analysis," in *4th International Workshop on Peer-to-Peer Systems (IPTPS)*, 2005. [Online]. Available: http://citeseer.ist.psu.edu/725723.html

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

[8] M. Shreedhar and G. Varghese, "Efficient fair queueing using deficit round robin." in *SIGCOMM*, 1995, pp. 231–242.

[9] A. R. Bharambe, C. Herley, and V. N. Padmanabhan, "Analyzing and improving a bittorrent networks performance mechanism," in *Proc. of IEEE Infocom*, Barcelona, Spain, 2006.

[10] A. Legout, N. Liogkas, E. Kohler, and L. Zhang, "Clustering and sharing incentives in bittorrent systems," in *Proc. of ACM SIGMETRICS 2007*, San Diego, CA, USA, June 2007.

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

[12] 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.

[13] T. B. Ma, C. M. Lee, J. C. S. Lui, and D. K. Yau, "Incentive and service differentiation in p2p networks: A game theoretic approach," *IEEE/ACM Transactions on Networking*, vol. 14(5), 2006.

[14] D. Erman, D. Ilie, and A. Popescu, "Bittorrent session characteristics and models," in *HET-NETs '05, Third International Working Conference on Performance Modelling and Evaluation of Heterogeneous Networks*, Ilkley, United Kingdom, 2005.

[15] M. Izal, G. Urvoy-Keller, E. E. Biersack, P. Felber, A. A. Hamra, and L.Garces-Erice, "Dissecting bittorrent: Five months in a torrents lifetime," in *PAM*, Antibes Juan-les-Pins, France, Apr 2004.

[16] J. A. Pouwelse, P. Garbacki, D. H. J. Epema, and H. J. Sips, "The bittorrent p2p file-sharing system: Measurements and analysis," in *4th International Workshop on Peer-to-Peer Systems*, Ithaca, NY, USA, Feb 2005.

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

[18] A. Legout, G. Urvoy-Keller, and P. Michiardi, "Rarest first and choke algorithms are enough," in *Proc. of ACM SIGCOMM/USENIX IMC'2006*, October 2006.

[19] R. Bindal and P. Cao, "Can self-organizing p2p file distribution provide qos guarantees?" in *OSR Special Issue on Self-Organizing Systems*, 2006.

[20] X. Yang and G. de Veciana, "Service capacity of peer to peer networks," in *Proc. of IEEE INFOCOM*, 2004.

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

[22] L. Massoulie and M. Vojnovic, "Coupon replication systems," in *Proc. ACM SIGMETRICS*, 2005.

[23] M. Lin, B. Fan, J. C. S. Lui, and D.-M. Chiu, "Stochastic analysis of file-swaring systems," *Performance Evaluation*, vol. 64(9-12), pp. 856–875, 2007.

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

[25] Y. Zhou, D. M. Chiu, and J. C. S. Lui, "A simple model for analysis and design of P2P streaming algorithms," in *IEEE ICNP*, 2007.

[26] Y. Huang, T. T. J. Fu, D. M. Chiu, J. C. S. Lui, and C. Huang, "Challenges, design and analysis of a large-scale p2p vod systems," in *ACM SIGCOMM*, 2008.

[27] 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*, vol. 63, pp. 175–194, March 2006.

[28] 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.

[29] M. Meo and F. Milan, "A rational model for service rate allocation in peer-to-peer," in *Proc. of IEEE INFOCOM*, 2005.

[30] E. Adar, "Drawing crowds and bit welfare," *SIGecom Exch.*, vol. 5, no. 4, pp. 31–40, 2005.

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

[32] M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy, and A. Venkataramani, "Do incentives build robustness in bittorrent?" in *Proc. of NSDI*, Cambridge, MA, USA, Apr 2007.

Dah-Ming Chiu received the B.Sc degree in Electrical Engineering from Imperial Collage, University of London, and the Ph.D degree from Harvard University, in 1975 and 1980 respectively.

He was a Member of Technical Staff with Bell Labs from 1979 to 1980. From 1980 to 1996, he was a Principal Engineer, and later a Consulting Engineer at Digital Equipment Corporation. From 1996 to 2002, he was with Sun Microsystems Research Labs. Currently, he is a professor in the Department of Information Engineering in The Chinese University of Hong Kong. He is known for his contribution in studying network congestion control as a resource allocation problem, the fairness index, and analyzing a distributed algorithm (AIMD) that became the basis for the congestion control algorithm in the Internet. His current research interests include economic issues in networking, P2P networks, network traffic monitoring and analysis, and resource allocation and congestion control for the Internet with expanding services. Two recent papers he co-authored with students have won best student paper awards from the IFIP Performance Conference and the IEEE NOMS Conference. Recently, Dr Chiu has served on the TPC of IEEE Infocom, IWQoS and various other conferences. He is a member of the editorial board of the IEEE/ACM Transactions on Networking, and the International Journal of Communication Systems (Wiley).

Bin Fan received the B.S. degree in Computer Science from University of Science and Technology of China, and the M.Phil degree in Computer Science from the Chinese University of Hong Kong. Currently he is a Ph.D. student in Computer Science Department at Carnegie Mellon University. He is a student member of IEEE.
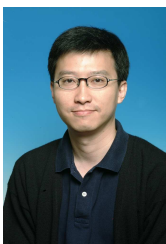
John C.S. Lui received his Ph.D. in Computer Science from UCLA. He is currently the chairman of the Computer Science & Engineering Department at the Chinese University of Hong Kong. His research interests span both in systems as well as in theory/mathematics with the emphasis on the robustness, scalability, and security issues on the Internet. John received various departmental teaching awards and the CUHK Vice-Chancellor's Exemplary Teaching Award, as well as the co-recipient of the Best Student Paper Awards in the IFIP WG 7.3 Performance 2005 and the IEEE/IFIP Network Operations and Management (NOMS) Conference. He is an associate editor in the Performance Evaluation Journal, IEEE-TC, IEEE-TPDS and IEEE/ACM Transactions on Networking. John was the TPC co-chair of ACM Sigmetrics 2005 and the General Co-chair for ICNP 2007.