BitTorrent Under a Microscope: Towards Static QoS Provision in Dynamic Peer-to-Peer Networks

Tom H. Luan and Xuemin (Sherman) Shen
Department of Electrical and Computer Engineering
University of Waterloo
Waterloo, ON N2L 3G1, Canada
Email: {hluan, xshen}@bbcr.uwaterloo.ca

Danny H. K. Tsang
Department of Electronic and Computer Engineering
Hong Kong University of Science and Technology
Hong Kong
Email: eetsang@ee.ust.hk

***Abstract*—For peer-to-peer (P2P) networks continually to flourish, QoS provision is critical. However, the P2P networks are notoriously dynamic and heterogeneous. As a result, QoS provision in P2P networks is a challenging task with nodes of the varying and intermittent throughput. This raises a fundamental problem: is stable and delicate QoS provision achievable in the highly dynamic and heterogeneous P2P networks?**

In this work, we investigate BitTorrent (BT) with the particular interest in its QoS performance in the highly dynamic and heterogeneous network. Our contributions are two-fold. First, we develop an analytical model to examine a randomly selected BT node under a microscope. Based on the model, we study the mean and variance of nodal download rate in the dynamic network and the performance of BT in QoS provision under different levels of peer churns. Our analysis unveils that although BT strives to provide nodes with guaranteed throughput, due to the network dynamics, the download rates of the peers oscillate extraordinarily and can hardly converge to the target QoS as proposed in previous literature. Second, to improve the QoS provision, we propose an enhanced protocol incorporating with BT. The proposed protocol enables nodes to quickly and elaborately search their uploaders, and as a result, achieve guaranteed and stable QoS in the dynamic networks. Using both analysis and simulations, we validate the effectiveness of the proposed protocol in comparisons with the original BT.

I. Introduction

The BitTorrent (BT)-like P2P content distribution networks currently represent the most promising driving wheel for large-scale content delivery over the Internet, engrossing nearly 35% of all Internet traffic [1]. However, the pervasive adoption of BT and its variants in a variety of applications, such as live [2] and on-demand video streaming [3], is arguable due to the lack of sufficient QoS support. In P2P networks, users download from peer nodes which are diverse in bandwidth and are susceptible to leave at any time. As a result, the download rate of nodes is inevitably intermittent and dramatically changing all the time, which directly throttles the performance of the on-top applications. More importantly, P2P networks rely on nodes to contribute their bandwidth. To encourage uploading, BT strives to ensure fairness [4] as the goal of QoS: nodes achieve download rate proportional to their upload rate[1]. Without effective QoS provision, peers will not be spurred to contribute, which may make the whole system corrupted. Therefore, *to understand and continually improve the QoS performance of BT in the dynamic P2P networks is critical*.

In the literature, several analytical models have been developed to evaluate the system performance of BT in terms of scalability and stability. Based on a branching process, Yang *et al.* [5] show that the service capacity of BT grows exponentially when a flash crowd of nodes arrives, indicating the resilience and scalability of the protocol. Qiu *et al.* [6] evaluate the BT-like network using a fluid flow model and show that the average download time of nodes is unrelated to the network size, which confirms the scalability of BT. However, both studies stem from a macroscopic view by evaluating the network-wide performance; the download performance of individual nodes is, however, neglected. Fan *et al.* [7] dissect the BT protocol and target on the QoS issue of BT. Using a static optimization model, they show that the resultant download rate of individual peers could be delicately controlled by the built-in QoS mechanisms of BT when the network converges to certain stabilized state. In other words, BT protocol is effective in QoS provision. However, Bindal *et al.* [8] reach the opposite conclusions using real-world experiments. [8] measures the download performance of two nodes with equal upload rate and observes that the download time of the two nodes is almost random and differs from each other significantly in all the trials, suggesting that BT can not provide guaranteed QoS to nodes. While there could be many potential causes, we argue that the deviation between [7] and [8] is mainly due to the network dynamics. The target QoS described in [7] is achieved in certain converged state. However, consistently churned by the network dynamics, such converged state may never be achievable in practice, making the QoS of BT invalid as reported in [8].

Insight of this, an immediate question is *how network dynamics affect the QoS of BT and how to provision stable QoS immune from the network dynamic and heterogeneity*. In this work, we provide a theoretical study on addressing this question. Unlike most previous work, we model BT from a microscopic view by focusing on a randomly selected node. We model the evolution of download connections of nodes using a Markov model and investigate the impacts of network dynamics on the connectivity of the node. We show that the download rate of nodes varies significantly due to the peer churns and can hardly converge to the steady QoS described in [7]. We argue that the poor performance of BT in the dynamic environment is due to the inefficient

[1]Throughout the paper, the target of QoS provision is to enforce the (proportional) fairness principle specified in [4], i.e., to provide nodes with the stable download rate matching their upload rate.

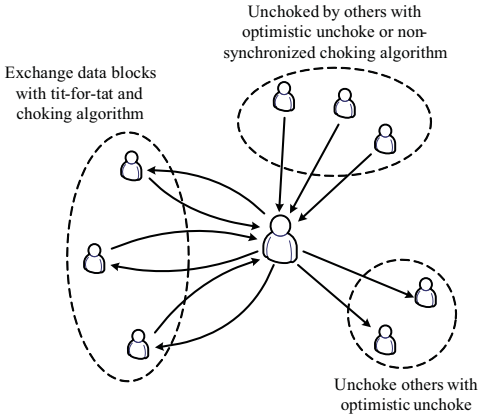

Fig. 1. Connections of nodes using the BT protocol suite

peer search scheme; by blindly connecting to peers, nodes in BT can hardly find qualified peers to download from. To remedy this, we propose an efficient peer search scheme incorporating with BT. By enabling peers to quickly and accurately locate the high-rate upload peers, our proposed protocol is able to provision guaranteed and stable QoS in the dynamic P2P network.

The remainder of this paper is organized as follows. Section II provides a description of the BT protocol and discusses on the QoS mechanisms of BT. Section III models the BT protocol from the perspective of a single node and evaluate its download performance using a Markov model. Section IV validates the correctness of the proposed model using simulations and evaluates the impact of network dynamics on the QoS performance of BT. In Section V, we devise an enhanced protocol incorporating with BT towards improved QoS, and validate its performance via both analysis and simulations. Section VI surveys the related works and Section VII closes the paper with conclusions.

II. Background

A. Description of BT

To accommodate the network dynamic and bandwidth heterogeneity, BT enforces the QoS using the iterative peer selection scheme which is composed of three mechanisms, namely *tit-for-tat*, *choking algorithm* and *optimistic unchoke*.

The tit-for-tat mechanism specifies that at any time instant a node only "unchoke" (BT term, means upload to) those which also unchoke it. As such, nodes exchange the downloaded data among each other using bidirectional connections. The goal of tit-for-tat is to forbid the free-riders – nodes only download from others without uploading.

The choking algorithm is to help nodes always exchange data blocks with the high rate peers. It proceeds as follows: at each node, the time is slotted into periodic cycles with each of T_c seconds ($T_c = 10$ by default). At the end of each cycle, a node ranks the peers uploading to it according to their upload rates and select a number of n_c peers which provide it the best upload rates to unchoke. (By default, $n_c = 4$.) As such, the choking algorithm is performed at individual peers in a distributed and non-synchronized manner with the goal to preserve high rate upload connections.

The optimistic unchoke works complementarily with the choking algorithm. In specific, every T_o seconds ($T_o = 3T_c$ by default), a peer randomly selects a number of n_o nodes to unchoke ($n_o = 1$ by default), even though those nodes are not uploading to it. As such, each node keeps $(n_c + n_o)$ upload connections at each time. The optimistic unchoke violates tit-for-tat, aiming to help nodes explore the high capacity peers to exchange data with. By unchoking peers for free, a node is expected to be reciprocated by the others when they perform the choking algorithm.

Fig. 1 illustrates the connections of a peer node using the above three mechanisms. Specified by the choking algorithm and optimistic unchoke, the upload links of nodes are constantly $(n_c + n_o)$. The number of download links of nodes, however, is random due to the optimistic unchoke from others and the non-synchronized choking algorithm among nodes.

B. QoS Provision through Clustering

It is broadly recognized that BT relies on the clustering behavior of nodes to provision QoS (i.e., ensure the fairness principle). In specific, by iteratively filtering peers based on their rates, nodes with the similar upload rate tend to form clusters to exchange data blocks, henceforth called cluster peers. The reason is as follows. If a node is connected in the higher capacity clusters, it will be "choked" (disconnected) frequently due to the relatively low upload rate and forced to leave the cluster. On the other hand, if it currently exchanges data in lower capacity clusters, it will keep choking others until finding a satisfactory high capacity node to exchange data with. Therefore, it is stable for nodes to exchange data in their own clusters. As such, the fairness is ensured as nodes upload and download at the similar rate in the cluster.

However, on the effectiveness of cluster formation and QoS provision in the dynamic network, there is a debate in existing literature. In [9], using experiments on the Planetlab, Legout *et al.* show the existence of clusters in a network composed of 40 peers and its effectiveness of QoS in BT. From a game theoretical point of view, Fan *et al.* [7] show that to form clusters and download from cluster peers is a Nash equilibrium as nodes achieve the best download rate in this case. Moreover, they show that with perfect clusters formed, the QoS of peers can be delicately controlled by modifying n_c and n_o. However, using experiments, Dale *et al.* [10] show that the clustering effect is not obvious when the network is large scale and extraordinarily heterogeneous. Based on this observation, they propose to confine the network size by allowing each node to communicate with a subset of peers only. Bindal *et al.* [8] also question on the effectiveness of the QoS mechanisms in BT and study based on a real-world measurement. They set up two separate peers on one machine to download the same video file through the same group of peers. As the two peers have the same status and are working in exactly the same environment, according to the fairness principle, they should download at the same rate. However, in all the 13 separate measurements, they observe that the finishing time of the two

peers differs in several tens of hours, which suggests that the download rate of the two peers are significantly different. Via a close investigation, they find that each node downloads 90% of the content from a small set of "close" peers. If one of its "close" peers departs from the network, a node needs to consume a very long time, typically over half an hour, to find another "close" peer to replace the departing one. As their "close" peers fail in a purely random and unpredicted fashion, the finishing time of the two peers is almost random. The observation in [8], on one hand, confirms the existence of clusters in BT in terms of "close" peers. On the other hand, it indicates that in practice the formation of the cluster is slow and fragile which could be easily churned by the node dynamics.

In this paper, we analytically examine the formation of cluster and effectiveness of QoS provision in BT in the highly dynamic environment. To this end, we provide a theoretical study on the download rate of individual peers with different settings of BT parameters and peer churn levels.

III. Analysis of QoS with Network Dynamics

In this section, we model the download connections of a randomly tagged node as a Markov process and study the clustering effect in the dynamic network by showing the composition and statistics of the download connections of the tagged peer.

A. Mathematical Model

We assume that peers arrive at the network in an uncoordinated and unpredictable fashion, following the Poisson distribution with the mean rate λ peers/second. The duration of time that a peer stays in the network is independently and exponentially distributed with parameter μ. We consider a stable network in which the network has proceeded for a relatively long period of time with the stabilized network size. Let N denote the mean network size. We have $\lambda = N\mu$ according to the Little's law. For ease of exposition, the network is composed of two classes of nodes: the high bandwidth (H-BW) nodes and the low bandwidth (L-BW) nodes. Let c_H and c_L denote the upload capacity of the H-BW nodes and L-BW nodes, respectively, and $c_H > c_L$. Let p_H and p_L denote the portion of H-BW peers and L-BW peers in the network, respectively, and $p_H + p_L = 1$. Throughout the work, we assume that the upload links are the bottlenecks. Such an assumption is fairly typical in the literature, such as [7], [11].

We focus on the achieved download rate of a single node using BT. To this end, we model the download connections of a randomly tagged node as a Markov process. Specifically, at each time t, the download connections of the tagged node is denoted by $(X(t), Y(t))$ where $X(t)$ represents the number of H-BW download connections[2] of the tagged node at time t and $Y(t)$ represents the number of L-BW download connections.

At time t, the download rate of the tagged node is thus

$$d(t) = X(t)\frac{c_H}{n_c + n_o} + Y(t)\frac{c_L}{n_c + n_o}, \tag{1}$$

[2]Throughout the work, the download (upload) connection refers to the connection via which the tagged node download from (upload to) others.

Here, using TCP connections (default in BT), we assume that peers evenly allocate their bandwidth over the concurrent upload connections. This is a working assumption also made in [11], [12].

Let $\mathcal{X}$ denote the class which the tagged node belongs to, where $\mathcal{X} \in \{\text{H-BW}, \text{L-BW}\}$. Let $\pi_{\mathcal{X}}(x, y)$ denote the steady state probability of the tagged node in state $X(t) = x, Y(t) = y$ when $t \to \infty$. The mean and variance of the download rate of the tagged node are, respectively,

$$\widehat{d}_{\mathcal{X}} = \lim_{t\to\infty} \sum_{x=0}^{p_H N} \sum_{y=0}^{p_L N} \pi_{\mathcal{X}}(x, y) d(t), \tag{2}$$

$$v_{\mathcal{X}} = \lim_{t\to\infty} \sum_{x=0}^{p_H N} \sum_{y=0}^{p_L N} \pi_{\mathcal{X}}(x, y) \left(d(t) - \widehat{d}_{\mathcal{X}}\right)^2. \tag{3}$$

For each node in the network, the mean download rate represents its long term QoS, while the variance of download rate represents the short term download performance. Given the file size, with a large variance of download rate, the file download time could vary significantly [8]. In what follows, we solve the Markov model in perspective of the tagged node and evaluate its mean and variance of download rates churned by the node dynamics.

B. Transition Rates

The non-null transition rates of the tagged node's download connections from state (x, y) to other states are shown in (4), where $q_{\mathcal{X}}(\cdot|\cdot)$ denotes the one-step transition rate. $\mathcal{CP}_{\text{A}\to\text{B}}$ is called the choking probability which represents the probability that a class A node chokes a class B node in the execution of the choking algorithm, where A, B $\in \{\text{H-BW}, \text{L-BW}\}$.

Eq. (4a) accounts for the rate at which the tagged node achieves a new download connection from the H-BW peers. This event comprises of three components: the first term in the RHS (right-hand side) of (4a) accounts for the rate at which the newly added download connection is due to the optimistic unchoke. This is because that there are on average $p_H N$ H-BW peers in the network. With each H-BW peer generating n_o optimistic unchoke links at the mean rate $1/T_o$, collectively H-BW peers issue optimistic unchoke at the mean rate $p_H N n_o/T_o$. As there are totally N nodes sharing the optimistic unchoke links, the tagged node is connected and increases one H-BW download connection at the rate $p_H n_o/T_o$. The second term in the RHS of (4a) accounts for the rate at which the tagged node is connected by a newly arrived H-BW node. This is because that the H-BW nodes arrive at the mean rate of $p_H\lambda$ nodes/second. With each arrival issuing $(n_c + n_o)$ upload connections which are shared by N nodes with equal probability, the rate at which the tagged node is connected is $p_H\lambda(n_c + n_o)/N$. As $\mu = \lambda/N$, the term can be simplified as $p_H\mu(n_c + n_o)$. The third term in the RHS of (4a) accounts for the rate at which the tagged node is unchoked by the choking algorithm of H-BW nodes. In specific, the tagged node randomly issues n_o optimistic unchoke links to the network for hunting H-BW nodes. Among them, on average $p_H n_o$ links are connected to H-BW nodes. Each H-BW node performs the

$$q_{\mathcal{X}}(x+1,y|x,y) = \frac{p_H n_o}{T_o} + p_H\mu(n_c+n_o) + \frac{1}{T_c}n_o p_H\left(1-\mathcal{CP}_{\text{H-BW}\to\mathcal{X}}\right), \tag{4a}$$

$$q_{\mathcal{X}}(x,y+1|x,y) = \frac{p_L n_o}{T_o} + p_L\mu(n_c+n_o) + \frac{1}{T_c}n_o p_L\left(1-\mathcal{CP}_{\text{L-BW}\to\mathcal{X}}\right), \tag{4b}$$

$$q_{\mathcal{X}}(x-1,y|x,y) = \begin{cases} \mu x + \frac{1}{T_c}x\mathcal{CP}_{\text{H-BW}\to\mathcal{X}}, & x < n_c, \\ \frac{1}{T_o}(x-n_c) + \mu x + \frac{1}{T_c}n_c\mathcal{CP}_{\text{H-BW}\to\mathcal{X}}, & x \geq n_c, \end{cases} \tag{4c}$$

$$q_{\mathcal{X}}(x,y-1|x,y) = \begin{cases} \frac{1}{T_o}y + \mu y, & x > n_c, \\ \frac{1}{T_o}(x+y-n_c) + \mu y + \frac{1}{T_c}(n_c-x)\mathcal{CP}_{\text{L-BW}\to\mathcal{X}}, & x \leq n_c < x+y, \\ \mu y + \frac{1}{T_c}y\mathcal{CP}_{\text{L-BW}\to\mathcal{X}}, & x+y \leq n_c. \end{cases} \tag{4d}$$

choking algorithm at the mean rate $1/T_c$, and at each round with probability $(1-\mathcal{CP}_{\text{H-BW}\to\mathcal{X}})$ that the connected H-BW node will unchoke the tagged node reciprocally. Expressions of the choking probability $\mathcal{CP}_{\text{H-BW}\to\mathcal{X}}$ will be derived later.

Eq. (4b) accounts for the rate at which the tagged node adds one L-BW download connection. It can be derived in a similar manner as (4a).

Eq. (4c) accounts for the rate at which the tagged node loses one H-BW download connection. In state (x,y), the tagged node is concurrently downloading from x H-BW peers. Among them, at most n_c nodes could be unchoked reciprocally by the tagged node using the choking algorithm. Therefore, the transition rate in (4c) differentiates according to x, as:

- ▷ When $x < n_c$, the tagged node will unchoke all the H-BW nodes uploading to it. The first term, μx, is the rate at which the H-BW nodes exchange data with the tagged node departing from the network, making the H-BW download connections of the tagged node decrease by one. The second term, $\frac{1}{T_c}x\mathcal{CP}_{\text{H-BW}\to\mathcal{X}}$, is due to the choking algorithm. This is because that in state (x,y) the tagged node unchokes x H-BW nodes concurrently, and each of them may choke the tagged node with the probability $\mathcal{CP}_{\text{H-BW}\to\mathcal{X}}$ in the iteration of choking algorithm at the rate $1/T_c$.
- ▷ When $x \geq n_c$, $(x-n_c)$ H-BW nodes will be choked by the tagged node. Those H-BW nodes unchoke the tagged node mainly with the optimistic unchoke and will choke the tagged node at the rate $1/T_o$ per second. The second term, μx, is the disconnection rate due to the departure of the H-BW upload nodes. The third term is due to the choking algorithm.

Eq. (4d) accounts for the rate at which the tagged node loses a L-BW download connection. It could be derived in a similar fashion as (4c).

C. Choking probability $\mathcal{CP}$

To solve the Markov model, we identify the choking probability $\mathcal{CP}$ for different classes of nodes.

Suppose that the tagged node is a H-BW node, the probability that it is choked by a class A node, where $\text{A} \in \{\text{H-BW}, \text{L-BW}\}$, is

$$\mathcal{CP}_{\text{A}\to\text{H-BW}} = \sum_{x=n_c+1}^{p_H N}\left(\sum_{y=0}^{p_L N}\pi_A(x,y)\frac{x-n_c}{x}\right). \tag{5}$$

The rational is as follows. Suppose that the class A node is in state (x,y). It will only choke H-BW nodes when $x \geq n_c+1$. In this case, as the class A node can only unchoke n_c nodes at most, it will choke $(x-n_c)$ H-BW nodes randomly selected from the x H-BW uploaders. The probability that the tagged node is selected is $(x-n_c)/x$. $\pi_A(x,y)$ is the probability that the class A node is in state (x,y).

If the tagged node is a L-BW node, the probability that it is choked by a class A node, where $\text{A} \in \{\text{H-BW}, \text{L-BW}\}$ is,

$$\begin{aligned}\mathcal{CP}_{\text{A}\to\text{L-BW}} &= \sum_{x=n_c}^{p_H N}\sum_{y=0}^{p_L N}\pi_A(x,y) \\ &\quad + \sum_{x=0}^{n_c-1}\sum_{y=n_c+1-x}^{p_L N}\pi_A(x,y)\left(1-\frac{n_c-x}{y}\right).\end{aligned} \tag{6}$$

This is because that a node in state (x,y) will choke a L-BW node with probability one if it has more than or equal to n_c H-BW uploaders, or $x \geq n_c$. If $x < n_c$, the node will unchoke all the x H-BW uploaders and randomly unchoke (n_c-x) nodes from the y L-BW uploaders. The probability that the tagged node is not selected to unchoke is $1-(n_c-x)/y$.

Given the transition rate matrix $\mathbf{Q}_{\mathcal{X}}$ with each non-null element $q_{\mathcal{X}}(\cdot|\cdot)$ shown in (4) and the choking probability $\mathcal{CP}$ shown in (5) and (6), the steady probability $\pi_{\mathcal{X}}$ of class $\mathcal{X}$ nodes, where $\mathcal{X} \in \{\text{H-BW}, \text{L-BW}\}$, can be derived with the following balance equations

$$\begin{cases} \boldsymbol{\pi}_{\mathcal{X}}\mathbf{Q}_{\mathcal{X}} = \mathbf{0}, \\ \sum_{x=0}^{p_H N}\sum_{y=0}^{p_L N}\pi_{\mathcal{X}}(n,k) = 1. \end{cases} \tag{7}$$

Eq. (7) is a self-contained non-linear system which could be solved using numerical methods.

The steady probability $\pi_{\mathcal{X}}(x,y)$ represents the clustering effect of BT. For perfect clustering, we should have node downloading from cluster peers only, i.e., $\sum_{y=0}^{p_L N}\pi_{H-BW}(n_c,y) = \sum_{x=0}^{p_H N}\pi_{L-BW}(x,n_c) = 1$. However, churned by the dynamic peer traffic and BT protocol itself[3], perfect clustering can hardly be achieved and the resultant distribution of download connections can be identified by our model. By substituting $\pi_{\mathcal{X}}(x,y)$ and (1) into (2) and (3), we can evaluate the download rates

[3]The iterative choking algorithm and optimistic unchoke further churn the network.

TABLE I
DEFAULT SETTINGS OF THE SIMULATOR

Network		BT Protocol				Node Bandwidth (kbps)		
λ	N	n_c	T_c	n_o	T_o	H-BW	L-BW	p_H
1	1000	4	10	1	30	1024	256	0.2

of peers. Given λ and μ, the BT parameters can be optimized towards the maximal fairness as

$$\begin{array}{cc} \max\limits_{n_c,n_o,T_c,T_o} & \sum_{\mathcal{X}} \log \frac{\widehat{d}_{\mathcal{X}}}{c_{\mathcal{X}}} \\ s.t., & v_{\mathcal{X}} \leq \xi, \end{array} \tag{8}$$

where ξ is a predefined value and $\mathcal{X} \in \{\text{H-BW}, \text{L-BW}\}$.

IV. SIMULATION EVALUATION

In this section, we conduct a simulation study to validate our analytical model in Section III and examine the QoS and clustering effect of BT in different network environments.

Our simulation is conducted using a session-level, event-driven simulator coded in C++. In each simulation run, there are 5000 peer arrivals, following the Poisson process with a mean rate λ (peers/second). Each peer arrival is associated an exponentially distributed life time with a mean $1/\mu$ (seconds); once the life time is over, the peer departs from the network. In all the simulation experiments, the mean network size N is kept constant with the mean overall peer arrival rate λ equal to the mean overall peer departure rate $N\mu$. We simulate the two-class network and compare the simulation results with analysis. The default parameter settings of the simulator are shown in Table I.

In the following simulations, we focus on a typical peer of the network and evaluate its QoS performance (mean and variance of download rate) in depth by: 1) changing the BT protocol parameters, i.e., n_c, n_o, T_c and T_o, and 2) adjusting the network dynamics with fixed BT parameters. To adjust the network dynamics, we keep the average peer population (N) to be 1000 and modify λ to achieve different peer churn levels in terms of peer arrivals and departures ($\mu = \lambda/N$) per unit time. For evaluation purpose, the selected peer is inserted after 1000 nodes join the network and is kept alive without leaving. In each experiment, we conduct two sets of simulations by assigning the selected node as a H-BW node and L-BW node, respectively. The experimental results are averaged over 30 individual simulation runs and reported with the 95% confidence interval.

Fig. 2 shows the download rate of the selected node over time in a typical experiment run. As we can see, while BT can differentiate the throughput of nodes according to their upload capacity, the download rate of nodes oscillates extraordinarily over time. With limited life time of nodes, their QoS can hardly be guaranteed.

Fig. 3 shows the mean download rate of the tagged node as a function of n_c. As described by Fan's model in [7], increasing n_c will improve the fairness of BT in that the download rate of nodes approaches to their upload rate. However, as can be seen in Fig. 3, Fan's model overestimates the QoS of nodes. This is because that [7] assumes a converged network with perfect clustering. Churned by dynamic peers, the mean download rate of nodes can hardly approach to the optimal value in [7]. Our model is accurate to estimate the mean download rate as it takes the peer churns into account. Fig. 4 depicts the variance of the download rate with increasing n_c. As we can see, increasing n_c could reduce the variance of download rate significantly and provision relatively stable download throughput to peers.

Fig. 5 shows the mean download rate as a function of n_o. As we can see, increasing n_o will decrease the fairness of BT which is because that peers spend more bandwidth to randomly unchoke others. However, the curve reduces with a smaller slope as Fan's model in Fig. 5. This is because that Fan's model neglects the interplay between n_c and n_o. When n_o increases, nodes attain enhanced ability to locate cluster peers, which boosts their performance accordingly in both the mean throughput and the variance of download rate as shown in Fig. 6.

Fig. 7 shows the mean download rate when increasing T_c and T_o, i.e., reducing the frequency of choking algorithm and optimistic unchoke. In this case, the fairness of nodes reduces. This is because that with increased T_c, nodes are slow to filter low rate peers, and therefore, are less capable of locating cluster peers. As a result, nodes can hardly exploit stable downloads within clusters and hence encounter much more dynamic download rates, as shown in Fig. 8. Note that overly reducing T_c and T_o would even churn the network, they should be optimally selected to balance the mean throughput and dynamics of the throughput via (8).

Fig. 9 shows the mean download rate when λ increases. In this case, the network becomes more dynamic with more peer arrivals and departures in a unit time, and the clusters formed among peer become more fragile. As a result, the fairness of nodes degrades with enhanced peer churns and deviates from Fan's model. The variance of download rate also increases as shown in Fig. 10 with the intensive peer churns.

In summary, in this section, we show that using the BT protocol, the download rate of peers are intensively churned and can hardly converge to the target value specified in [7]. By taking the network dynamics into account, our model is more accurate to understand of QoS performance in different levels of network dynamics.

V. ENHANCED BT TOWARDS IMPROVED QOS

Although there are a number of potential causes, we conjecture that the main cause of the imperfect clustering and accordingly the inaccurate QoS provision of BT is due to the inefficient search scheme used in the optimistic unchoke. In specific, to locate the cluster nodes, the optimistic unchoke randomly unchoke peers and explore cluster peers using a trial-and-error process. In this process, if a node of relatively high bandwidth is unchoked, the node may either not reciprocate or reciprocate temporally only as it expects to exchange data with its own cluster peers. On the other hand, if a node of lower bandwidth is unchoked by the optimistic unchoke, the node would reciprocate immediately to preserve the high rate connection, and choke its cluster peer instead. However, the node may be choked soon when the optimistic unchoke expires. It then needs to find another

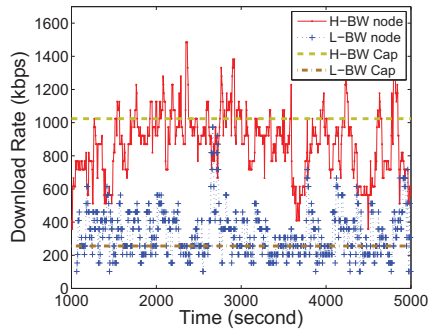

Fig. 2. Download rate of the tagged peer over time

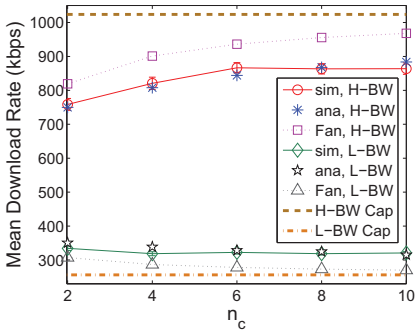

Fig. 3. Mean download rate of the tagged peer with increasing n_c

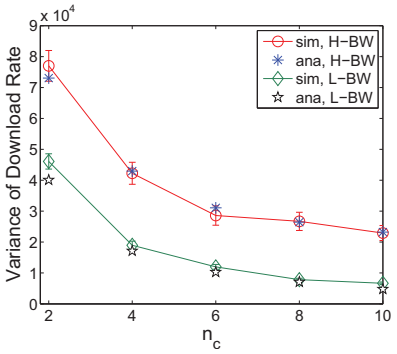

Fig. 4. Variance of the download rate of the tagged peer with increasing n_c

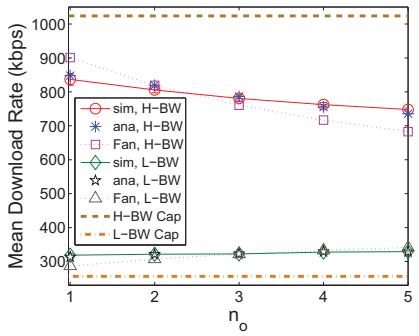

Fig. 5. Mean download rate of the tagged peer with increasing n_o

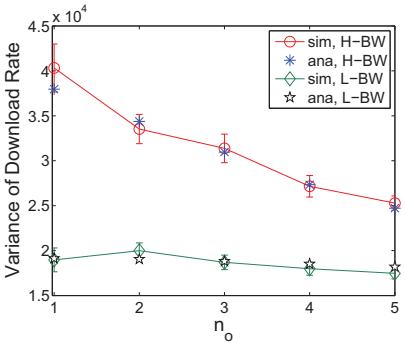

Fig. 6. Variance of the download rate of the tagged peer with increasing n_o

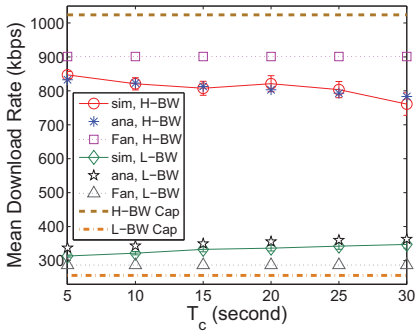

Fig. 7. Mean download rate of the tagged peer with increasing T_c and $T_o = 3T_c$

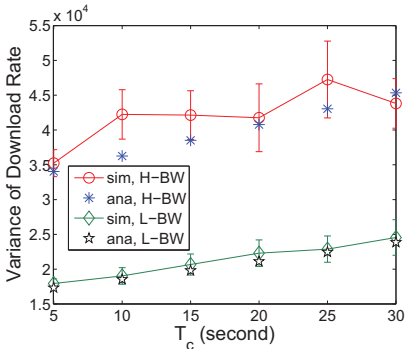

Fig. 8. Variance of the download rate of the tagged peer with increasing T_c and $T_o = 3T_c$

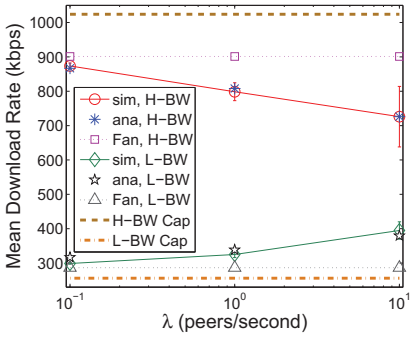

Fig. 9. Mean download rate of the tagged peer with increasing λ

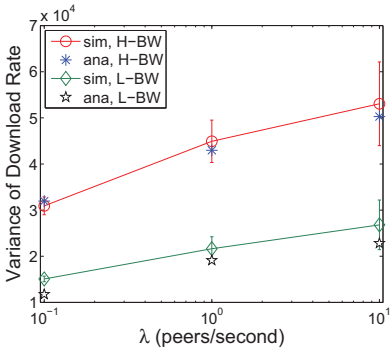

Fig. 10. Variance of the download rate of the tagged peer with increasing λ

cluster peer again, making its download unstable. In this point of view, the optimistic unchoke is harmful to the formation of low bandwidth clusters. More importantly, it is very difficult to find the cluster nodes by blindly unchoking others in the peer ocean, especially when the network is highly dynamic and heterogeneous. As reported in [8], the time of locating a cluster node is extraordinarily long in practice, and the clustering effect is very weak in a highly dynamic and heterogeneous network [10].

To remedy this, in this section, we propose a more intelligent peer search scheme to replace the random trial-and-error search used in the optimistic unchoke. The proposed scheme is built upon the idea of forming link-level homogeneous networks. In specific, while nodes have diverse upload rates, by tuning the number of their upload connections in proportion to the upload rate, we can make nodes have equal upload rates upon each upload connection, namely link-level homogeneity. In this case, there is no need to differentiate peers to download from, or equivalently, all nodes now belong to the same cluster. Moreover, when losing a cluster peer to exchange data with, a node could soon connect to another one to replace, making the QoS immune from the network dynamics. In what follows, we present the

proposed node search scheme in details towards the formation of link-level homogeneity.

A. Proposed Search Scheme

The proposed search scheme is based on a random walk algorithm (inspired by [13]) as follows. To search for appropriate nodes via the optimistic unchoke, instead of randomly unchoking others, a peer first issues multiple random walkers to the network with each walker forwarded among peers in a fully distributed manner. At each intermediate node, e.g., i, the walker is forwarded to the next hop probabilistically as

$$p_{ij} = \begin{cases} \frac{1}{|\mathcal{N}_i(t)|+1} \min\left\{\frac{c_j^2 k_i(t)(k_i(t)+1)}{c_i^2 k_j(t)(k_j(t)+1)}, 1\right\}, & j \in \mathcal{N}_i(t), \\ 1 - \sum_{j\in\mathcal{N}_i(t)} p_{ij}, & j = i, \end{cases} \tag{9}$$

where $\mathcal{N}_i(t)$ denotes the set of neighbor peers of node i. The neighbor peers of a node are maintained by the built-in mechanisms of BT and will be explained later. They are connected to node i for collecting node information only without any data block exchange. $|\mathcal{N}_i(t)|$ represents the cardinality of $\mathcal{N}_i(t)$. c_i and c_j denote the upload capacity of node i and its neighbor peer j, respectively. $k_i(t)$ and $k_j(t)$ denotes the number of upload connections of node i and j at time t, respectively.[4]

Each walker traverses TTL (Time-to-Live) hops among peers. The last node receiving the walker is selected to be unchoked by the peer issuing the walker.

According to the Metropolis-Hastings algorithm, it can be shown that by forwarding walkers with (9), a node, e.g., j, in the network will be unchoked with the probability

$$\pi_j(t) = \frac{\frac{c_j^2}{k_j(t)}}{\sum_{i\in V(t)} \frac{c_i^2}{k_i(t)}}, \tag{10}$$

where $V(t)$ denotes the set of nodes in the network. In this manner, a node with larger capacity and smaller number of upload connections will be unchoked with the higher probability.

B. Enhanced BT Protocol

The enhanced BT protocol incorporated with the proposed random walk based optimistic unchoke is as follows.

1) Join Phase: When a new node joins the network, it first connects to the tracker[5] and downloads a list of nodes which join the network previously from the tracker. This set of nodes constitute the initial set of neighbor peers $\mathcal{N}(t)$ of the arrived node. As nodes are dynamically leaving the network, a peer communicates with the tracker whenever the number of its neighbor peers is below a threshold, denoted by n_{min}. At each time, a node fetches a list of n_{max} neighbor nodes at most.

Upon its arrival, each node selects m peers (e.g., $m = n_c + n_o$) to unchoke. Instead of randomly unchoking others as in the original BT, the node issues m walkers and selectively unchokes the node using the random walk algorithm as described in the previous subsection. Whenever a peer is unchoked in this process, it will reciprocate to unchoke the newly arrived node immediately and therefore establishes a bidirectional connection with the newly arrived node for data exchange. As a result, each arrival establishes m connections with bidirectional data exchange in the join phase.

2) Download Phase: During the download period, a node performs the choking algorithm iteratively at the interval of T_c seconds. Same as the original BT, a node chokes the download nodes with the smallest upload rate to it and rebuilds a new link to replace using the random walk algorithm. Whenever a node, e.g., i, is choked by another, either in the execution of choking algorithm of others or due to the departure of a upload node, it performs the random walk based optimistic unchoke and rebuilds one link with the rebuilding probability

$$r_i = \begin{cases} 1, & k_i = 2, \\ r, & k_i > 0, \end{cases} \tag{11}$$

where r is a predefined value and $0 < r \leq \frac{(N-1-2m)\mu T_c}{2+(N-1)\mu T_c}$. The upper bound of r is derived in [14]. After a new peer is selected in this phase, a bidirectional connection is established between this peer and i for data exchange.

As a result, the enhanced BT protocol maintains an undirected mesh topology all the time with bidirectional uploads of data along each connection. As such, free-riders are banned in the same principle of tit-for-tat. In the next, we show that in such a topology, the network will converge to the link-level homogeneity, i.e., all connections coverage to have equal bandwidth. As the data rate is equal along each direction of the link, the fairness of nodes can be guaranteed in the dynamic network.

C. Proof of Link-level Homogeneity and QoS

We use the same notations defined in the previous sections. Assuming that peers are perfectly selected with the probability in (10), the number of upload connections $k_j(t)$ of a randomly selected node j evolves over time t with

$$\begin{aligned} \frac{dk_j(t)}{dt} &= \left(\lambda m + \mu\widehat{k}(t) r + \frac{N}{T_c}(1+r)\right)\pi_j(t) \\ &\quad -\mu k_j(t)(1-r) - \frac{k_j(t)}{\widehat{k}(t) T_c}(1-r), \end{aligned} \tag{12}$$

where $\widehat{k}(t)$ is the average number of upload connections of each node at time t. Asymptotically, we have

$$\lim_{t\to\infty} \widehat{k}(t) = \frac{\mu m T_c + r}{\mu(1-r) T_c}. \tag{13}$$

The detailed derivation of (13) is shown in [14].

The first term on the RHS of (12) accounts for the rate at which $k_j(t)$ increases. In this term, λm accounts for the generation rate of random walkers by the new arrivals, as nodes arrive at the rate λ and each arrival issues m walkers. $\mu\widehat{k}(t) r$ is the rate of walkers generated by the departing peers. This is because that nodes depart from the network at the rate μ and each departure causes on average $\widehat{k}(t)$ nodes to issue walkers to reconnect to a new peer with the probability r in (11). $\frac{N}{T_c}(1+r)$ is the rate of

[4]The number of upload connections in the proposed scheme is proportional to the upload rate of nodes and is no longer fixed to $(n_c + n_o)$ as in the original BT.

[5]The tracker is commonly used in BT to bootstrap the newly arrived nodes.

walkers generated in the choking algorithm of nodes collectively. In specific, there are N nodes in the network on average and each node periodically selects one node to choke in the choking algorithm of the enhanced BT at the rate $1/T_c$. In this process, the node which chokes others will issue a new random walker to connect to another nodes as in the enhanced BT, while the node which is choked also issues a walker with probability r to reconnect to another peer. Of all the walkers generated in a unit time, node j is selected with probability $\pi_j(t)$, making $k_j(t)$ increase.

The second and third terms of the RHS of (12) collectively amount to the rate at which $k_j(t)$ decreases. The second term is due to the departure of nodes who are exchanging data with node j. At time t, node j exchanges data with $k_j(t)$ nodes. Each of them departs at the rate μ and with probability $(1-r)$, node j will not reconnect to any others to replace the lost connection. The third term is due to the choking algorithm of others. Specifically, among the nodes exchanging data with node j, they each has on average $\widehat{k}(t)$ upload connections and periodically performs the choking algorithm at the rate $1/T_c$. Assuming that with equal probability node j will be choked. Once choked, with probability $1-r$, node j will not reconnect to others to rebuild the download connection.

Solving (12) in the steady state when $k_j(t)$ converges with $\frac{dk_j(t)}{dt}=0$, we have

$$\lim_{t\to\infty}\frac{c_j}{k_j(t)}=\frac{\widehat{c}\lambda P}{\Omega}, \tag{14}$$

where $\widehat{c}$ is the average capacity of nodes. Ω and P are constants as

$$\Omega=\frac{\lambda m+\mu\widehat{k}r+\frac{N}{T_c}(1+r)}{(1-r)\left(\mu+\frac{1}{\widehat{k}T_c}\right)}, \tag{15}$$

$$P=\lim_{\tau\to\infty}\int_0^{\tau}\frac{e^{-\mu\tau}}{\sqrt{1-e^{-2Q\tau}}}d\tau, \tag{16}$$

with $Q=(1-r)\left(\mu+\frac{1}{\widehat{k}T_c}\right)$. Refer to [14] for the derivation of (14).

Assuming that nodes equally allocate their capacity over each upload connection, (14) indicates that after a long enough time the upload connections of nodes coverage to have equal bandwidth. In this case, as each connection in the enhanced BT topology is bidirectional, nodes have download rate equal to their upload rate, which is the fairness pursued in BT.

D. Comparison with the Original BT

In this part, we compare the enhanced BT protocol with the original BT protocol using simulations. Similar to Section IV, we examine a selected peer, which is inserted to the network when 1000 node has joined, using both original BT and enhanced BT protocols. The setting of the original BT network is same to Section IV, except that we simulate a more heterogeneous network for both protocols. The capacity c of each node is selected with probability $Pr(c)=0.1\delta(c-256)+0.3\delta(c-512)+0.45\delta(c-1024)+0.15\delta(c-2048)$, where $\delta(x)=1$ if $x=0$, and $\delta(x)=0$, otherwise. In the enhanced BT network, we set $m=5, TTL=10$ for the random walk algorithm, and $r=0.2$, $n_{min}=10$ peers, $n_{max}=40$ peers, $T_c=10$ seconds.

Fig. 11 plots the download rate of nodes over time. As can be seen, using the original BT protocol, the download rate oscillates extraordinarily, while the download rate of nodes using the enhanced BT protocol is very stable and nearly equal to the upload capacity.

Fig. 12 shows the capacity per upload connection of nodes when $t=4000$ seconds. As we can see, around 70% of peers have the capacity per upload connection converged to the analytical value in (14), which demonstrates the link-level homogeneity in the network.

Finally, we simulate the case when the capacity of the selected peer changes every 1000 seconds and plot the download rate in Fig. 13. In real world situations, the upload capacity of nodes may change from time to time because the bandwidth is shared among multiple applications besides BT. In this scenario, the download rate should adapt to the varying upload capacity. However, as we can see in Fig. 13, using the original BT protocol, the download rate of nodes nearly remains the same and is very slow to chase the change of upload capacity. In contrast, using enhanced BT protocol, the download rate can effectively adapt to the varying upload capacity even with the severe peer churns. In a nutshell, our proposed enhanced BT could provision stable and delicate QoS control adaptive to the upload rate.

VI. Related Works

This section briefly reviews the previous literature and highlights our contributions in light of existing works.

A prevalent approach for evaluating the BT performance is by using the fluid model. [6] proposes the first fluid model. By assuming homogeneous capacity and full bandwidth utilization, [6] shows that the network is scalable with the mean download time of nodes unrelated to the network size. [15] extends [6] by considering a heterogeneous network. Similar to our work, [15] also studies the clustering effect and sketches the download rate of nodes. However, it stands from a macroscopic view of the network without modeling the details of the protocol. [16] evaluates the completeness of file download in the dynamic network and shows that the fraction of nodes in different download stages is a U-shape curve. [2] implements the fluid flow model for analyzing BT-like live streaming applications. While the fluid flow model is simple, it normally assumes the global knowledge of the network and evaluates the network-wide performance. Neglecting the protocol details, it fails to unveil the QoS of specific peers, which is the concern of our work.

Another representative approach is by using the static and probabilistic model. [7] focuses on the tradeoff between the fairness and download rate of nodes and shows that the number of choking and optimistic unchoke links corresponds to important tuning parameters in striking the balance between the download rate and fairness. As the conclusions of [7] rely on the assumption of perfect clustering, [11] extends the model by considering the imperfect clustering and shows the download rates of nodes with the existence of seed nodes[6] and free-riders. [12] also models BT

[6]Nodes finish downloading and upload only

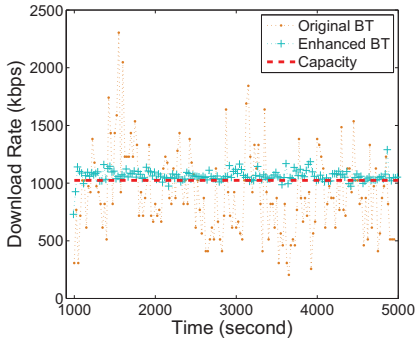

Fig. 11. Download rate of the selected peer over time

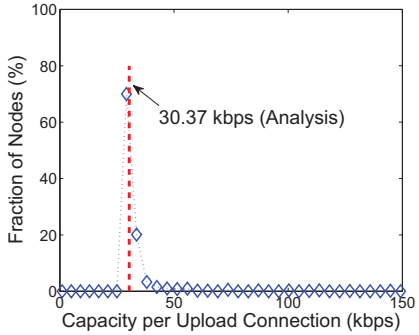

Fig. 12. Distribution of capacity per upload connection of nodes at time $t = 4000$ seconds

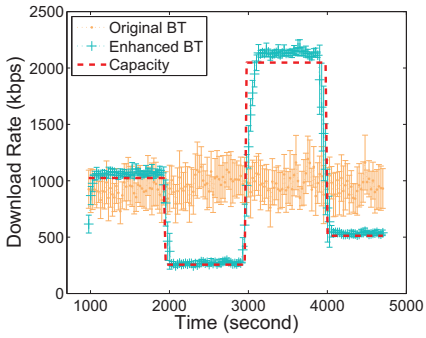

Fig. 13. Download rate over time with the selected peer adapting upload capacity every 1000 seconds

in a heterogeneous network based on a probabilistic model. In general, the static model focuses on the static network and can not study the impacts of network dynamics on the QoS. In contrast, our work explicitly models the dynamic node traffics and peer churns. For this reason, our model can unveil the interplay between BT and network environment and provide insights on the QoS and clustering of BT in the dynamic environment. Unlike the static model which can only show the mean performance in the long term run, we are able to study the short term QoS in terms of variance of downloading. With limited life time, we argue that the short term QoS is more important to nodes and deserves elaborate study.

In parallel to the theoretical studies, the impacts of network dynamics on the clustering and QoS of nodes are evaluated extensively in experimental studies. [9] pinpoints the existence of clustering effects using Planetlab experiments. However, even in a small scale closed system, their results indicate that the clustering of nodes is still far from perfect. Using the real-world experiments, [17] finds that 80% of BT nodes receive more data than upload, severely violating the fairness of BT. To reinforce clustering, [10] proposes to make use of the tracker to help cluster peers upon their arrivals and [18] also introduces an protocol to replace the optimistic unchoke towards more robust and efficient clustering. While extensively studied using measurements, the clustering and QoS of BT remain unmodeled in a heterogeneous and dynamic scenario. Along this direction, to the best of our knowledge, our work represents the first research effort.

VII. Conclusion

We conclude this paper by stressing that ensuring stable and elaborate QoS guarantee is crucial to BT-like P2P networks. In an effort to address this issue, we have provided a mathematical model to evaluate the QoS performance achieved by individual nodes in the highly dynamic networks. Due to the peer churns, we have shown that the clusters formed among nodes are fragile which results in highly dynamic and instable download performance to nodes. To remedy that and improve the QoS in dynamic networks, we have also proposed an enhanced peer search scheme to incorporate with the original BT. Through both analysis and simulations, we have demonstrated that using the enhanced BT protocol the download rates of nodes converge fast and accurately to the desired QoS value and are resilient to the network dynamics. Encouraged by our results in this paper, we intend to work towards a real-world deployment of the enhanced BT as the future work.

References

[1] Ernesto, "BitTorrent: the "one third of all Internet traffic" myth," http://torrentfreak.com.

[2] S. Tewari and L. Kleinrock, "Analytical model for bittorrent-based live video streaming," in *Proc. of IEEE CCNC*, 2007.

[3] Y. Yang, A. L. Chow, L. Golubchik, and D. Bragg, "Improving QoS in BitTorrent-like VoD Systems," in *Proc. of IEEE Infocom*, 2010.

[4] B. Cohen, "Incentives build robustness in BitTorrent," in *Proc. of ACM P2PECON*, 2003.

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

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

[7] B. Fan, J. C. Lui, and D.-M. Chiu, "The design trade-offs of BitTorrent-like file sharing protocols," *IEEE/ACM Transactions on Networking*, vol. 17, no. 2, pp. 365–376, 2009.

[8] R. Bindal and P. Cao, "Can self-organizing P2P file distribution provide QoS guarantees?" *ACM Sigops Operating Systems Review*, vol. 40, no. 3, p. 30, 2006.

[9] A. Legout, N. Liogkas, E. Kohler, and L. Zhang, "Clustering and sharing incentives in BitTorrent systems," in *Proc. of ACM Sigmetrics*, 2007.

[10] C. Dale, J. Liu, J. Peters, and B. Li, "Evolution and Enhancement of BitTorrent Network Topologies," in *Proc. of IEEE IWQoS*, 2008.

[11] A. L. Chow, L. Golubchik, and V. Misra, "BitTorrent: an extensible heterogeneous model," in *Proc. of IEEE Infocom*, 2009.

[12] W.-C. Liao, F. Papadopoulos, and K. Psounis, "Modelling BitTorrent-like systems in heterogeneous environments," *IEEE Transactions on Parallel and Distributed Systems*, submitted.

[13] K. Kwong and H. Tsang, "Building heterogeneous peer-to-peer networks: protocol and analysis," *IEEE/ACM Transactions on Networking (TON)*, vol. 16, no. 2, pp. 281–292, 2008.

[14] T. H. Luan, X. Shen, and D. H. Tsang, "BitTorrent Under a Microscope: Towards Static QoS Provision in Dynamic Peer-to-Peer Networks," BBCR, University of Waterloo, Tech. Rep., 2010.

[15] M. Meulpolder, J. Pouwelse, D. Epema, and H. Sips, "Modeling and analysis of bandwidth-inhomogeneous swarms in BitTorrent," in *Proc. of IEEE P2P*, 2009.

[16] Y. Tian, D. Wu, , and K. W. Ng, "Modeling, analysis and improvement for bittorrent-like file sharing networks," in *Proc. of IEEE Infocom*, 2006.

[17] M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy, and A. Venkataramani, "Do incentives build robustness in BitTorrent," in *Proc. of USENIX NSDI*, 2007.

[18] R. Izhak-Ratzin., "Collaboration in BitTorrent systems," in *Proc. of IFIP Networking*, 2009.