Stochastic Analysis and File Availability Enhancement for BT-like File Sharing Systems[*]

Fan Bin   Dah-Ming Chiu   John C.S. Lui

Abstract

In this paper, we present the mathematical analysis of two important performance measures for a BitTorrent (BT) like P2P file sharing system, namely, average file downloading time and file availability. For the file downloading time, we develop a model using the "*stochastic differential equation*" approach, not only it captures the system more *accurately* than some previous approach [18], but also allows us to capture various network settings and peers behavior. We study the steady-state behavior and obtain the closed-form solutions for performance measures such as the average number of peers, the average system throughput, average file downloading time. These analytical results allow us to carry *sensitivity analysis* on various performance measures for various system parameters. We then extend this model to consider multiclass peers wherein some peers are behind firewalls which may impede the uploading service. We also present the mathematical model to study the file availability of a BT-like system. The model helps us gain the understanding of why the "*rarest-first*" chunk selection policy is used in today's BT protocol. We show under some situations this policy may not be good in practice and propose a novel chunk selection algorithm to enhance the overall system file availability. Extensive simulations are carried to validate our analysis.

Keywords: Peer-to-peer, BitTorrent, Modeling, Performance evaluation, File availability

1 Introduction

For the past few years, peer-to-peer (P2P) file sharing systems are generating tremendous amount of traffic on today's Internet. This form of communication paradigm is reshaping the way new network applications are being designed. For example, one can find P2P softwares for multimedia file sharing (i.e., video and audio files), live video streaming applications [22], as well as distribution of software patches [11].

Compared with the traditional client/server paradigm, the P2P approach has a much better scalability property. Specifically, when one scales up the number of users, the performance such as the file downloading time for the client/server architecture can degrade substantially, while the P2P architecture has an attractive

[*] Fan Bin and John C.S. Lui are with the Computer Science & Engineering Department at the Chinese University of Hong Kong. Dah-Ming Chiu is with the Information Engineering Department at the Chinese University of Hong Kong.

property that more users can actually improve the file downloading performance. This property is especially true for the BitTorrent (BT) protocol [1]. Another interesting features of the BitTorrent protocol is the built-in incentive to share information, which encourage users to cooperate so files can be downloaded quickly.

The main contributions of our work are:

- We develop a fluid model for BT-like P2P systems based on the "*stochastic differential equation*" (SDE) technique [6], rather than the simple differential equation approach. The SDE approach allows us to obtain closed-form solution for the transient and the steady state performance measures such as number of downloaders, numbers of seeders, the average file downloading time. We show that our results are not only more accurate than the previous work [18], but it allows us to perform important sensitivity analysis of the performance measures on various system parameters such as file popularity, effect of seeders, connection probability,...,etc.
- We extend the above model to allow class differentiation. In particular, we consider a class of peers which are behind firewalls, which is common these days, and these peers may impede the uploading process of the overall system.
- We present the mathematical model for predicting the file availability in BT system. The model allows us to gain the understanding as to why the *rarest-first* policy is used as the built-in chunk selection algorithm in BT. We also present the rationale why this policy may *not* be optimal and we propose a more efficient chunk selection algorithm to enhance the file availability.
- Both analytical models are validated by a discrete event simulation which is detailed enough to capture many of BT's features[1]. These analytical results provide us the important insights for designing a BT-like protocol. Also, as compared with the simple fluid model in [18], not only our model is more accurate, but our model focuses more on characterizing details of heterogeneous peers with reasonable network topology and network parameters, and at the same time, maintains the model simplicity and mathematical tractability.

The balance of our paper is as follows. In Section 2, we provide a basic introduction to BitTorrent and a brief review of related work. In Section 3, we present the mathematical model to describe the dynamics of a BT-like P2P system as well as its performance measures. In Section 4, we extend this mathematical model to accommodate heterogeneous peers, i.e., some of the peers are behind firewalls. File availability model is presented in Section 5. Section 6 concludes.

2 Background and Previous Work

BitTorrent is a peer-to-peer application designed to facilitate file sharing among multiple peers across unreliable networks [1]. In BT-like systems, files are split into equal-sized segments which are called *chunks*

[1] Some of the previous research results did not perform model validation.

(the typical size of a chunk is 32 to 256 KB) so that peers can download different chunks from multiple peers concurrently. To download a file, one peer should first get a *torrent* file which contains the necessary information such as the chunk number, chunk size, checksum and the file *tracker*. A tracker is a node in a BT system which keeps track of all peers that are interested in downloading and sharing a particular file. Usually, the URL of a tracker is contained in the corresponding torrent file. A newly joined peer can contact the tracker and the tracker will return a subset of peers who are currently in the BT system, and these peers become the neighbors of this newly joined peer. Under a BT-like system, peers that are downloading and sharing chunks with other peers are called "*leechers*". After collecting all chunks of the intended file, peers may choose to stay in a BT system and upload chunks to other peers. Peers that have all chunks are called "*seeders*". Initially, a BT system has at least one seeder, which is the first peer that wants to share the intended file with others. Under the BitTorrent protocol, there is no specification as to how long a peer should stay as a seeder. In fact, a peer can choose to abort in the middle of the download, or choose to leave the system immediately after it gets all the necessary chunks.

There are two important features in the BT protocol, namely, the "*rarest-fist*" chunk selection policy and the "*tit-for-tat*" peer incentive policy [8]. Using the rarest-first policy, a leecher will download one of its missing chunks and that chunk is the rarest chunk found in all its connected peers. The objective of this mechanism is to enhance the overall file availability (we will justify the use of the rarest chunk policy in Section 5). The tit-for-tat policy is a mechanism which aims to prevent free-riding [15] so that peers who refuse to upload chunks to other peers may not receive any download service.

Let us briefly summarize the related work on this topic. Recently, there are a number of analytical and measurement-based studies of BT-like systems. In [13], authors present the measurement results collected during a five-month period that involves thousands of peers, and evaluate the performance of the algorithms and mechanisms used by BT. In [17], authors present an eight-month trace-based study and measurement results of the popularity and the availability of BT systems. In [20], authors analyze the measurement result collected by a modified client in a BT network and propose a P2P-based streaming protocol. In [3], authors study the ability of the BT protocol to disseminate very large files among peers and present measurement results over a duration of four months. In [4], authors conduct various simulation-based experiments to investigate the effect of network parameters and system settings on the performance of file downloading.

For the mathematical analysis aspect, authors in [9, 21] propose a coarse-grain Markovian model to represent a P2P file sharing system. However, this Markovian model cannot capture many important properties of a BT-like system. Furthermore, these is *no closed form solution* for the steady state performance measure and one can only use numerical method to calculate these measures. To overcome the computation problem in [9, 21], authors in [18] propose a fluid model and a set of differential equations to describe the dynamics of BT systems and discuss issues like incentive mechanisms and free-riding. Note that the model in [18] is not accurate in the performance prediction (we will illustrate this in later section), and also fails to capture many intrinsic and important properties of BT-like P2P systems such as node degree and number of file sharing connections. Also, these previous works do not consider the underlying overlay topology and treat the effective throughput of peers as a constant. In [16], authors develop a detailed Markovian model to investigate the scalability and effectiveness of a P2P system. However, the result is more of theoretical interest since the model has a huge state space and it is difficult to analyze. Instead, one has to reply on asymptotic analysis. In [7], authors extend the model in [18] to illustrate the performance issue of providing service differentiation in a BT-like system. Similar to [18] wherein many simplified assumptions are made and essential network parameters are omitted which impede fundamental understanding on BT systems. In [12], authors make some correction of the model of [18] and present a multi-torrent collaboration policy. In [2] authors model the distribution of the individual chunk under multiple network topologies and routing algorithms. As for availability measure, the authors in [11] experimentally show that by using the network coding scheme, the system is much more robust than the BitTorrent protocol in the extreme scenario where the original seeder leaves immediately after distributing few copies to the system.

3 Mathematical Model for BT Dynamics

To represent the dynamics and evolution of a BitTorrent-like P2P system, we use a fluid model with a simplified state space using the *stochastic differential equation approach* [6]. Performance measures such as the average number of leechers, the average number of seeders, the average file downloading time and the overall system throughput are derived.

3.1 Analytical Model

Consider a BitTorrent-like P2P system that distributes a given file $\mathcal{F}$ to a large number of cooperative peers. The file is divided into M orthogonal chunks such that $\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2 \cup \cdots \cup \mathcal{F}_M$, where $\mathcal{F}_i \cap \mathcal{F}_j = \emptyset$ for $i \neq j$ and $\mathcal{F}_i$ is the i^{th} chunk of the file. For simplicity of analysis, we assume no network coding or erasure code is applied in the file sharing process. Typically, the number of chunks M is in the order of thousands. Based on BT's definition, a *seeder* is a peer which has all M chunks of $\mathcal{F}$ while a *leecher* is a peer which only has a subset of $\mathcal{F}$. Assume at time t, there are $N(t)$ peers in the system. These peers want to obtain and share the file $\mathcal{F}$, and new peers arrive according to a Poisson arrival process with rate λ. By the help of a tracker, each peer maintains a connection with another peer as its neighbor with a connectivity probability $\rho \leq 1$. One can view the BT file sharing system as an overlay network and every node in the overlay network has an average degree of $\rho(N(t)-1) \approx \rho N(t)$. For each connection, the average downloading rate is μ. Each peer is constrained by a maximum transfer rate B, which includes the downloading and the uploading rates Although a peer can keep logical connections to many peers, a peer can have at most B/μ uploading and/or downloading connections simultaneously. After collecting all chunks of $\mathcal{F}$, a leecher becomes a seeder and may serve others by uploading chunks. A seeder can choose to leave a BT-like system and the average departure rate is γ (i.e., $1/\gamma$ is the average time a seeder stays in the BT system). We let $c_i \leq M$ to represent the number of chunks that peer i is holding.

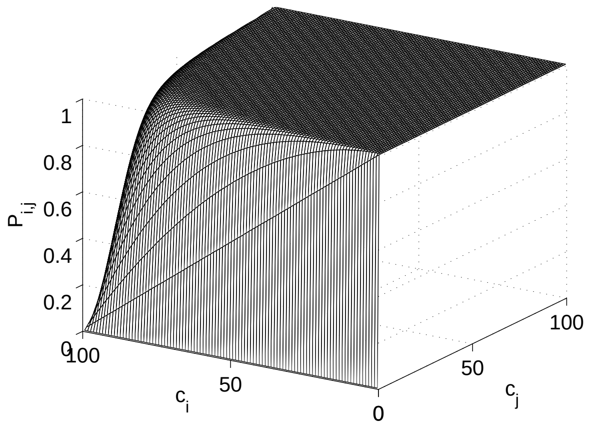


Figure 1: Probability $P_{i,j}$ when $M = 100$

In [18, 21], all peers are considered as having the same effectiveness η to contribute to the system. However it is not true in reality because when a new peer first enters the system, it has no chunk to upload. Even after some time it collects a small number of chunks, the effectiveness of this "new" peer is very different from peers with large number of chunks. On the other hand, if we consider all combination of different chunks [16] (i.e. peers with only $\mathcal{F}_1, \mathcal{F}_2$ and peers with only $\mathcal{F}_3, \mathcal{F}_4$ are of different types so there are 2^M states in the model), then the state space is extremely large. In this paper, we use a different approach and distinguish the states of peers by the number of chunks they are holding (i.e. peers with only $\mathcal{F}_1, \mathcal{F}_2$ and peers with only $\mathcal{F}_3, \mathcal{F}_4$ are of the same type so we need $M + 1$ states). Assume chunks are uniformly distributed among peers, which actually could be ensured by the *rarest-first* chunk selection policy. Let peer i and peer j have c_i and c_j chunks respectively, where $c_i, c_j \in \{0, 1, \dots, M\}$. Let us derive the probability that peer i can obtain *at least* one useful chunk from peer j, which we denote as $P_{i,j}$. When $c_i < c_j$, it is clear that $P_{i,j} = 1$. When $c_i \geq c_j$, we have:

$$\begin{aligned} P_{i,j} &= 1 - \text{P[chunks in peer } j \text{ are subset of chunks in peer } i\text{]} \\ &= 1 - \frac{\binom{c_i}{c_j}}{\binom{M}{c_j}} = 1 - \frac{c_i \cdot (c_i - 1) \cdots (c_i - c_j + 1)}{M \cdot (M-1) \cdots (M - c_j + 1)}. \end{aligned} \qquad (1)$$

So given the number of chunks peer i and peer j holding, we can estimate the probability $P_{i,j}$ (as illustrated in Figure 1). From Eq. (1), we need use $M + 1$ variables to capture the system dynamics. The problem is, the number of all states M is still a large number, can one reduce the number further? From Figure 1, one can observe that $P_{i,j}$ increases very sharply. So we use this important observation to reduce the state space.

We distinguish three types of peers: Type 1 peer is a leecher that holds a few chunks (i.e., say less than half of the M chunks). Type 2 peer is a leecher that holds most but not all chunks. Type 3 peer represents a seeder in the system. The probability $P_{i,j}$ in Eq. (1) can be simplified based on the following cases:

- **case 1:** If peer i is of type 1 or type 2, and peer j is of type 3, then clearly $P_{i,j} = 1$ since a seeder can always provide a useful chunk to a leecher.

- **case 2:** If peer i is of type 1 and peer j is of type 1 or type 2, then c_i/M is very small and we have $P_{i,j} \geq 1 - (c_i/M)^{c_j} \approx 1$.
- **case 3:** If peer i is of type 2 and peer j is of type 1, then c_i/M is close to 1 but since c_j is small, we have $P_{i,j} \approx 0$.
- **case 4:** If peer i and peer j are of type 2, then c_j is large and $(c_i/M)^{c_j} \approx 0$, so $P_{i,j} \approx 1$.

Now to represent the *heterogeneity* of peers' effectiveness while keeping the model simple and analytically tractable, we assign $P_{i,j}$ only two possible values: 0 or 1 according to the types of peer i and j.

Let $X_1(t)$, $X_2(t)$ and $Y(t)$ be the random variables representing the number of type-1 peers, type-2 peers and type-3 (seeders) in the system at time t. By case 1 and 2 of the analysis of Eq. (1), type-1, type-2 peers and seeders can assist type-1 peers in the file download process. Also, type-2 peers and seeders can assist type-2 peers based on case 1, 2 and 4 above. Let $D_i(t)$ and $U_i(t)$ denote the random variables of the downloading and uploading rates for $peer_i$ at time t. When there is *no bandwidth constraint* (i.e., B is infinitely large):

$$E[D_i(t)] = \begin{cases} \mu\rho\left(E[X_1(t)] + E[X_2(t)] + E[Y(t)]\right) & i \text{ is type-1} \\ \mu\rho\left(E[X_2(t)] + E[Y(t)]\right) & i \text{ is type-2.} \end{cases} \quad (2)$$

When we constrain a peer with bandwidth B, it means that for each peer i, the inequality $D_i(t) + U_i(t) \leq B$ needs to be satisfied. From the system's perspective, we have the following conservation rules:

$$\sum_{j=1}^{N(t)} D_j(t) + \sum_{j=1}^{N(t)} U_j(t) \leq BN(t), \quad (3)$$

$$\sum_{j=1}^{N(t)} D_j(t) = \sum_{j=1}^{N(t)} U_j(t). \quad (4)$$

Substitute Eq. (4) to Eq. (3) and taking the expectation. By the Wald's Equation [19], we have:

$$E[D_i(t)] \leq B/2. \quad (5)$$

Combining Eq. (2) and (5) and let $D^{(1)}(t)$ and $D^{(2)}(t)$ be the random variables denoting the downloading rate at time t for type-1 and type-2 peer respective, we have:

$$\begin{aligned} E[D^{(1)}(t)] &\approx \min\{\mu\rho(E[X_1(t)] + E[X_2(t)] + E[Y(t)]), B/2\} \\ E[D^{(2)}(t)] &\approx \min\{\mu\rho(E[X_2(t)] + E[Y(t)]), B/2\}. \end{aligned} \quad (6)$$

We can now present the mathematical model that captures the dynamics of a BT-like system. The model is based on the *stochastic differential equation* [6]. First, the arrival process of peers is modeled as a Poisson counter process $N(t)$ with an average arrival rate λ. The Poisson counter has the following properties:

$$dN(t) \; = \; \left\{ \begin{array}{ll} 1 & \text{at Poisson arrival} \\ 0 & \text{elsewhere,} \end{array} \right. , E[dN(t)] \; = \; \lambda dt. \tag{7}$$

Let $X_1(t)$ and $X_2(t)$ denote the number of type-1 and type-2 leechers at time t while $Y(t)$ denote the number of seeders in the system at time t. The following equations describe the *rate of change* of these three important variables:

$$\begin{array}{rcl} dX_1(t) & = & dN(t) - \frac{D^{(1)}(t)X_1(t)dt}{sM/2}, \\ dX_2(t) & = & \frac{D^{(1)}(t)X_1(t)dt}{sM/2} - \frac{D^{(2)}(t)X_2(t)dt}{sM/2}, \\ dY(t) & = & \frac{D^{(2)}(t)X_2(t)dt}{sM/2} - \gamma Y(t)dt. \end{array} \tag{8}$$

The rate of change of $X_1(t)$ is affected by the number of new arrival, which is denoted as $dN(t)$, and the number of peers that transfer from type-1 to type-2 is denoted by $\frac{D^{(1)}(t)X_1(t)dt}{sM/2}$, where $sM/2$ represents the size of a half of the file $\mathcal{F}$, and $D^{(1)}X_1(t)dt$ represents the amount of new information that all $X_1(t)$ type-1 peers collect in dt. Similarly, the transfer rate from type-2 peers to seeders is $\frac{D^{(2)}(t)X_2(t)dt}{sM/2}$. Lastly, since the departure rate of a seeder is γ, so the total departure rate of all seeders is represented by $\gamma Y(t)$. Taking the expectation of Eq. (8), we have:

$$\begin{array}{rcl} dE[X_1(t)] & \approx & E[dN(t)] - \frac{E[D^{(1)}(t)]E[X_1(t)]dt}{sM/2}, \\ dE[X_2(t)] & \approx & \frac{E[D^{(1)}(t)]E[X_1(t)]dt}{sM/2} - \frac{E[D^{(2)}(t)]E[X_2(t)]dt}{sM/2}, \\ dE[Y(t)] & \approx & \frac{E[D^{(2)}(t)]E[X_2(t)]dt}{sM/2} - \gamma E[Y(t)]dt. \end{array} \tag{9}$$

Note that the above equations are approximations because we are assuming the independence of $D^i(t)$ and $X_i(t)$, for $i = 1, 2$.

3.2 Steady-State Performance Measures

To study the steady-state performance, we let $dE[X_1(t)] = dE[X_2(t)] = dE[Y(t)] = 0$. To simplify notation further, we use $\bar{W}$ to represent the expected value of the random variable W and let $\alpha = \frac{2\mu\rho}{sM}$ and $\beta = \frac{B}{2\mu\rho}$ to simplify the expressions. To find the steady state solution, we classify Equation (9) into three cases:

Case 1 $\bar{X}_1 + \bar{X}_2 + \bar{Y} < \beta$,

Case 2 $\bar{X}_2 + \bar{Y} < \beta \leq \bar{X}_1 + \bar{X}_2 + \bar{Y}$, and

Case 3 $\beta \leq \bar{X}_2 + \bar{Y}$.

The first case implies that the uploading and downloading process are *not* constrained by the bandwidth B. This occurs when peers have broadband access to the Internet, or when the peer's arrival rate is low so there are only few peers in the system. For the second case, type-1 peers are constrained by bandwidth B while type-2 peers are not constrained by this bandwidth limit. The justification for this case is that there are more peers who can help type-1 peers than type-2 peers. Hence it is possible that former peers are saturated by the bandwidth constraint, yet not the latter. For the last case, all peers are constrained by the bandwidth B in the file sharing process. This case occurs when peers have a low bandwidth connection to the Internet, or the file is very popular so that the peer's arrival rate is very high and there are many peers in the system. We can solve $\bar{X}_1$, $\bar{X}_2$, $\bar{Y}$ respectively in these three cases. The following theorem below states the equilibrium point $\bar{\mathbf{X}} = (\bar{X}_1, \bar{X}_2, \bar{Y})$ of Eq. (9):

Theorem 1 (Equilibrium point) *In the regime $E[X_1(t)]$, $E[X_2(t)]$ and $E[Y(t)]$ are nonnegative, Eq. (9) has a unique equilibrium point $\bar{X}$:*

$$\bar{\mathbf{X}} = \begin{cases} \left(\frac{\sqrt{5}-1}{2}\sqrt{\frac{sM\lambda}{2\mu\rho}} - \frac{\lambda}{4\gamma}, \sqrt{\frac{sM\lambda}{2\mu\rho}} - \frac{\lambda}{2\gamma}, \frac{\lambda}{\gamma}\right) & \textit{if } \frac{1+\sqrt{5}}{2}\sqrt{\frac{\lambda}{\alpha}} + \frac{\lambda}{4\gamma} < \beta \quad \textit{(for Case 1)}, \\ \left(\frac{sM\lambda}{B}, \sqrt{\frac{sM\lambda}{2\mu\rho}} - \frac{\lambda}{2\gamma}, \frac{\lambda}{\gamma}\right) & \textit{if } \sqrt{\frac{\lambda}{\alpha}} + \frac{\lambda}{2\gamma} < \beta \leq \frac{1+\sqrt{5}}{2}\sqrt{\frac{\lambda}{\alpha}} + \frac{\lambda}{4\gamma} \quad \textit{(for Case 2)}, \\ \left(\frac{sM\lambda}{B}, \frac{sM\lambda}{B}, \frac{\lambda}{\gamma}\right) & \textit{if } 0 < \beta \leq \sqrt{\frac{\lambda}{\alpha}} + \frac{\lambda}{2\gamma} \quad \textit{(for Case 3)} \end{cases} \tag{10}$$

Proof: Due to the lack of space, we refer our readers to the technical report [10]. ∎

Theorem 2 (Local Stability) *The equilibrium point given by Theorem 1 is asymptotically stable.*

Proof: Due to the lack of space, we refer our readers to the technical report [10]. ∎

Theorem 3 *Let $\bar{T}_d$ denote the average downloading time for the file $\mathcal{F}$, which is the average time it takes for a peer to obtain all M unique chunks of $\mathcal{F}$. We have the following results:*

$$\bar{T}_d = \begin{cases} \frac{1+\sqrt{5}}{2}\sqrt{\frac{sM}{2\mu\rho\lambda}} - \frac{3}{4\gamma} & \textit{Case 1}, \\ \sqrt{\frac{sM}{2\mu\rho\lambda}} + \frac{sM}{B} - \frac{1}{2\gamma} & \textit{Case 2}, \\ \frac{2sM}{B} & \textit{Case 3}. \end{cases} \tag{11}$$

Proof: By the Little's result [14], $\bar{T}_d$ is given by $\bar{T}_d = \frac{\bar{X}_1 + \bar{X}_2}{\lambda}$. By Theorem 1, we can obtain the above results easily. ∎

Theorem 4 *Let $\bar{T}_p$ denote the average system throughput of the BT-like P2P system, the average number of peers in the system is $\bar{N} = \bar{X}_1 + \bar{X}_2 + \bar{Y}$. We have the following result:*

$$\bar{T}_p = \begin{cases} O(\bar{N}^2) & \textit{Case 1}, \\ O(\bar{N}) & \textit{Case 2 or 3}. \end{cases} \tag{12}$$

Proof: Due to the lack of space, we refer our readers to the technical report [10]. ∎

The above theorems provide the following *important insights*:

Remark 1: Quantifying the scalability of BitTorrent-like P2P networks: Based on the steady state system throughput as given by Eq. (12), one can find that the BT-like system scales well with the number of peers. Case 1 represents the system under a low arrival rate, therefore a small number of peers exists in the system. The throughput of the system is of the order of $O(\bar{N}^2)$. When there are more peers in the systems (i.e., in case 2 and 3), the system throughput is linearly proportional to the number of peers. So the system performance will *not* degrade as we scale up the number of peers.

Remark 2: Quantifying the sensitivity of downloading time to arrival rate: The intensity of the arrival rate represents the popularity of the file. To understand the impact of file popularity on the performance of BT-like P2P systems, we consider the rate of change of $\bar{T_d}$ when one increases the peer's arrival rate λ. Based on the expression of $\bar{T_d}$ in Eq. 12, we have:

$$\frac{\partial \bar{T_d}}{\partial \lambda} = \begin{cases} -\frac{1+\sqrt{5}}{4\sqrt{\alpha}}\lambda^{-3/2} & \text{Case 1,} \\ -\frac{1}{2\sqrt{\alpha}}\lambda^{-3/2} & \text{Case 2,} \\ 0 & \text{Case 3.} \end{cases}$$

For case 1 and 2, the average downloading time decreases when the arrival rate λ increases; in case 3, the rate of change of $\bar{T_d}$ is not related to λ. This means if the file is popular (i.e., large value of λ), the average downloading time will be smaller. Therefore the BT-like system scales well with the file popularity.

Remark 3: Quantifying the effect of the presence of seeders: Since γ represents the departure rate for seeders, $T_s = 1/\gamma$ is the average time a seeder stays in a P2P system. For case 1 and 2, when T_s increases, there will be more seeders in the system to provide the uploading service, therefore, the average downloading time $\bar{T_d}$ will decrease. Notice that

$$\frac{\partial \bar{T_d}}{\partial T_s} = \begin{cases} -3/4 & \text{Case 1,} \\ -1/2 & \text{Case 2,} \\ 0 & \text{Case 3.} \end{cases}$$

This implies that having more seeders will reduce the file downloading time. But when all peers are saturated due to the bandwidth limit, having more seeders will not improve the performance. Consider an extreme case of $T_s = 0$, that is, a peer will leave the system immediately after it downloads the entire file $\mathcal{F}$.

$$\lim_{\gamma \to \infty} \bar{T_d} = \begin{cases} \frac{1+\sqrt{5}}{2}\sqrt{\frac{sM}{2\mu\rho\lambda}} & \text{Case 1,} \\ \sqrt{\frac{sM}{2\mu\rho\lambda}} + \frac{sM}{B} & \text{Case 2,} \\ \frac{2sM}{B} & \text{Case 3.} \end{cases}$$

The above expression implies that peers can still obtain the file, though with higher downloading time, without the help of many seeders in the system.

Remark 4: Quantifying the effect of the connection probability ρ: A close examination of Eq. (11) reveals that $\bar{T}_d$ is a function of the connectivity parameter ρ for case 1 and 2 but not case 3. Increasing the value of ρ will reduce the value of $\bar{T}_d$. This is due to the fact that a peer has more neighbors to reduce its downloading time, as long as it is not saturated by its own bandwidth limit. In case 1 and case 2, increasing ρ will decrease $\bar{T}_d$, because larger ρ increases the possibility of downloading for peers. In case 3, ρ will not affect $\bar{T}_d$ because the system is operating at the saturated mode. One may think a larger value of ρ will always benefit a peer. However it is important to note that larger value of ρ will also cause peers to keep too many TCP connections. Hence a large value of ρ will increase the burden of the peers with too many connection overheads and eventually leads to saturating peers' bandwidth. Since ρ is affected by the number of peers reported by the tracker to a peer, a proper selection of this number is an interesting and practical problem.

Remark 5: Quantifying the effect of bandwidth constraint B: Consider the marginal utilization of B:

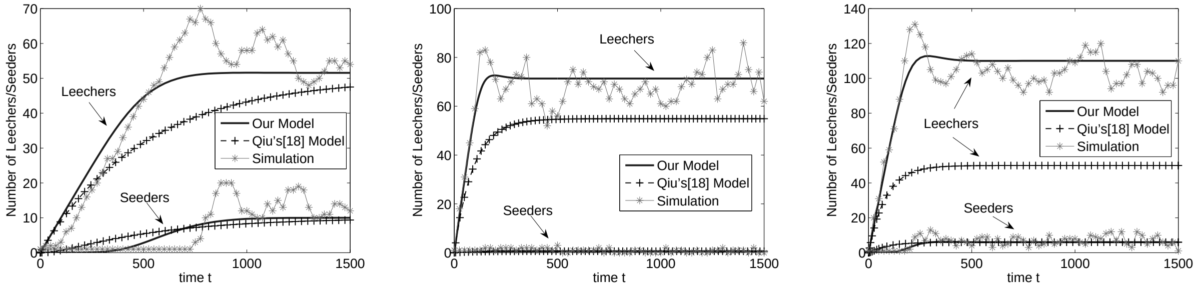
$$\frac{\partial \bar{T}_d}{\partial B} = \begin{cases} 0 & \text{Case 1,} \\ -\frac{sM}{B^2} & \text{Case 2,} \\ -\frac{2sM}{B^2} & \text{Case 3.} \end{cases}$$

For case 1, the bandwidth is not fully utilized so $\bar{T}_d$ is not affected by B, and more bandwidth is not helpful in this case. For case 2 and 3, by increasing the bandwidth limit, a peer can get a better performance. Given the above analysis, one can better anticipate the system's need since most BitTorrent implementations allow users to configure the maximum bandwidth.

3.3 Model Validation and Evaluation

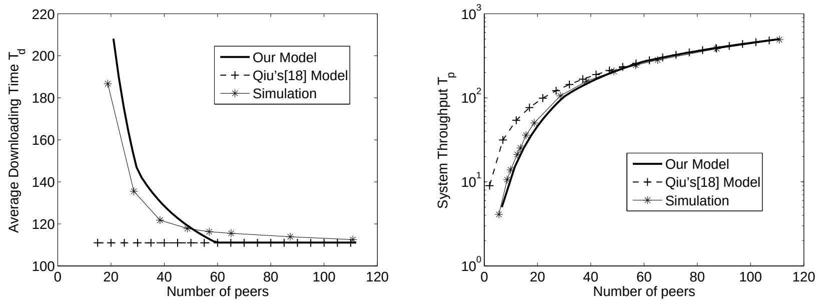
In this section, we perform a series of experiments to *validate* our analytical results. First, we implement a discrete event simulator for a BitTorrent-like file sharing system. The input of the simulator are parameters such as arrival rate, transfer rate between peers, departure rate of seeds, connection probability, transmission bandwidth of peers, etc. Our simulator models the behaviors of peers such as joining the system, making connections to neighboring nodes, selecting chunks for download, transfer chunks, updating the chunk bitmaps, seeding and also departures of seeders.

Experiment. 1 (Accuracy in estimating number of peers): In the following experiments, we consider the accuracy of the proposed mathematical model in estimating $E[X_1(t)]$, $E[X_2(t)]$ and $E[Y(t)]$. We also use this experiment to test the accuracy of the [18]'s model. In Fig.2, we compare the average number of leechers ($E[X_1(t)] + E[X_2(t)]$) and the average number of seeders ($E[Y(t)]$) with the simulation results. Fig.2(a) illustrates the case that the peer's arrival rate is $\lambda = 0.1$, seeder's departure rate is $\gamma = 0.01$, the transfer rate is $\mu = 0.1$ between two peers, the maximum transfer bandwidth of a peer is $B = 2$ and the connection probability is $\rho = 0.25$. The setting represents the situation that peers with low download bandwidth, and the maximum transfer rate between peers is low. Because the peer's arrival rate is low, so the file is not that popular. One can see that our model can accurately track the dynamics of the leechers



(a) (b) (c)

Figure 2: Comparing dynamics of peer evolutions for our model and Qiu's model under three different cases

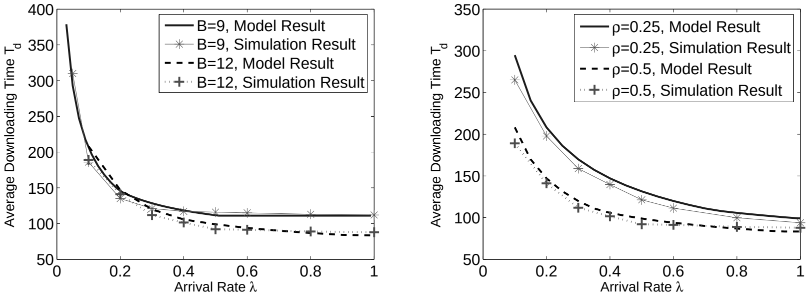


(a) $\bar{T}_d$ as the function of N (b) $\bar{T}_p$ as the function of N

Figure 3: Comparing System Scalability for our model and Qiu's model

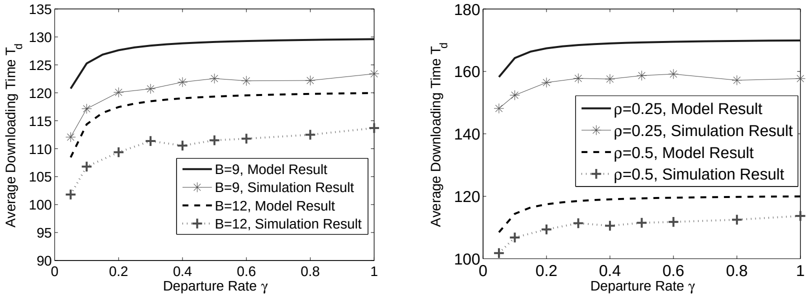
and seeders, while model based on [18] is only accurate in estimating the number of leechers and seeders in the steady state case. Fig.2(b) illustrates the case that the peer's arrival rate is $\lambda = 0.6$, seeder's departure rate $\gamma = 1.0$, peer's downloading bandwidth is $\mu = 0.3$, peer's maximum transfer bandwidth is $B = 12$ and the connection probability is $\rho = 0.25$. In this setting, the file is more popular so the peer's arrival rate is higher. Also, peers have a high downloading rate and a higher maximum transfer bandwidth. However, the seeder's departure rate is also higher than the previous experiment. Again, our model can accurately track the dynamics of the leechers and seeders, while model based on [18] *underestimates* the number of leechers in the system. Lastly, Fig.2(c) illustrates the case that the peer's arrival rate is $\lambda = 0.6$, seeder's departure rate $\gamma = 0.1$, downloading bandwidth between peers is $\mu = 0.3$, peer's maximum transfer bandwidth is $B = 12$ and the connection probability is $\rho = 0.1$. Note that our model can accurately track the dynamics of the leechers and seeders, while model based on [18] significantly *underestimates* the number of leechers in the system.

Experiment. 2 (Accuracy for Performance Measures $\bar{T}_d$ and $\bar{T}_p$): In this experiment, we investigate the accuracy of the derived performance measures, namely, the average downloading time $\bar{T}_d$ and system



(a) For $B = 9$ and 12 (b) For $\rho = 0.25$ and 0.5

Figure 4: $\bar{T}_d$ as the function of arrival rate λ



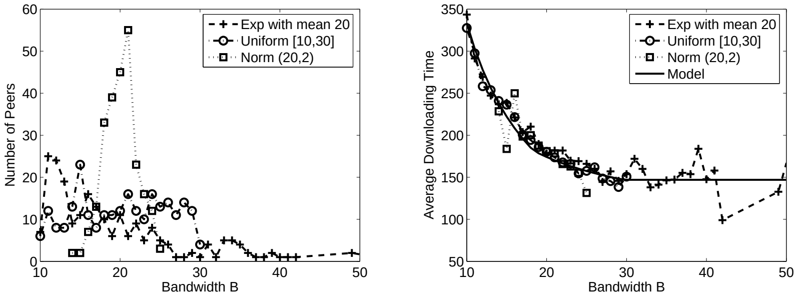
(a) For $B = 9$ and 12 (b) For $\rho = 0.25$ and 0.5

Figure 5: $\bar{T}_d$ as the function of departure rate γ

throughput $\bar{T}_p$. We set $M = 500$, $\mu = 0.3$, $\gamma = 1.0$, $\rho = 0.5$, $B = 9$ and vary the number of peers in the system. As shown in Fig.3, the BT-like system scales well with the number of peers. Note that our analytical results match well with the simulation results while Qiu's model underestimate (overestimate) $\bar{T}_d$ ($\bar{T}_p$). Also, there is a decrease of average downloading time when more peers are in the system. This property is also reported from the real BT-trace data [21]. The near linear relationship between the number of peers and the system throughput is reflected in our model and is also reported in [3].

Experiment. 3 (Sensitivity Analysis): In this set of experiments, we investigate the sensitivity of performance measures to various system parameters such as the arrival rate λ, the seeder's departure rate γ, the connection probability ρ and transmission bandwidth B.

3a) **The relationship between T_d and arrival rate λ:** For this experiment, we set M,μ and γ the same as in Experiment 2, but vary the arrival rate λ under different values of B and ρ. Fig.4(a) and 4(b) illustrate the effect on the average downloading time. Both of these figures show that when the value of arrival rate becomes large, the average downloading time decreases monotonically and eventually reaches a fixed value when the transmission bandwidth is saturated.



(a) Bandwidth VS Number of Peers

(b) Bandwidth VS Average downloading time

Figure 6: File downloading time vs. bandwidth under different bandwidth distributions

3b) **The relationship between T_d and departure rate γ:** In this experiment, we also set the parameters λ, M and μ the same as in Experiment 2 but now we vary the values of leaving rate γ. Fig.5(a) illustrates the average downloading time for $B = 9$ and 12 while Fig.5(b) illustrates the average downloading time for $\rho = 0.25$ and 0.5. These two figures also confirm that by increasing the departure rate γ, the seeder spends less time in the system, hence the average downloading time for peers increases. Notice that when γ is large enough, the rate of deterioration on the file downloading time approaches zero. This implies that even when there is no incentive for peer to be a seeder, the BT-like system can still provide service to peers in the system.

3c) **The relationship between T_d and connection probability** ρ. From Fig.4(b) and Fig.5(b), we observe that when there are more connections to peers (i.e., ρ is of high value), then the file downloading time actually decreases. From Fig.4(b), we observe that more highly connected system has a smaller downloading time, especially when λ is small. As λ increases, the performance difference between different values of ρ diminishes. So for a system with a low arrival rate, high connection probability of peers is important to improve the performance.

3d) **The relationship between T_d and bandwidth:** In Fig.5(a), the system with a higher bandwidth has a lower average downloading time. But in Fig.4(a), we can find that for the low arrival rate case, higher transfer bandwidth does not necessarily bring better performance. One can achieve better performance when the peer's arrival rate is high because there will be more peers contributing to the uploading process.

Experiment. 4 (Bandwidth Heterogeneity): The average downloading time given by Eq. (11) is derived under the assumption of homogenous bandwidth B for all peers in the system. In this experiment we relax this assumption and examine the case that peers join the system with different bandwidth. Still using parameters in Experiment 2 except B, we repeat the simulation using different bandwidth distribution, namely, (a) exponential distribution with mean 20, (b) uniform distribution in $[10, 30]$, and (c) normal distribution with mean 20 and variance 2. Fig. 6(a) shows the number of peers corresponding to the bandwidth in these three runs. The simulation results measured by average downloading time of the peers with certain bandwidth B is illustrated in Fig. 6(b). An interesting observation is that the downloading time of peers with a particular value of bandwidth is actually "independent" on the bandwidth distribution in all three runs. In other words, the average downloading time of a specific peer is mainly determined by its own bandwidth instead of the bandwidth of its neighbors. For example, for peers with bandwidth 20, the average downloading time is around 180, indepedent of the bandwidth distribution as normal, exponential, or uniformly distributed. And it is quite close to the model prediction given by Eq. (11) (setting $B = 20$) which is 174. Thus from Fig. 6(b), Eq (11) is a good performance predictor for the downloading time even when now peers have heterogenous bandwidth. Having this observation, we can use the analytical results we obtained to investigate the impact heterogenous peers in a BT-like system.

4 Model Extension For Peers behind Firewalls

In this section, we investigate the impact of firewall (or the network-address-translation box) on the BT protocol. Although recently some implementations of BitTorrent enable users behind different firewalls or NATs connected to each other via UDP, it still remains a problem for TCP. In general, a peer with a public IP address *cannot* initiate a TCP connection with a peer behind a firewall since the address of the latter peer is unknown. One way to establish a connection (both for the downloading and uploading of chunks) between these two different classes of peers is to involve a third party(i.e. the BT tracker). To illustrate, consider a peer a which is behind firewall while a peer b has a public IP address. When peer a joins the BT system, it has to contact the tracker so as to obtain a sublist of connecting peers. During this contact, the tracker remembers the "address" of peer a. When peer b joins the system, the tracker can inform peer a to initiate the connection with peer b (i.e. a peer behind the firewall needs to initiate the connection). In this way, a connection between peer a and b can be established. It is also important to note that when two peers are behind different firewalls (i.e. under different network domains), they cannot establish connection with each other since they do not know the "address" of each other. This implies that peers behind different firewalls cannot assist each other in the chunk uploading. This form of interaction is illustrated in Figure 7 wherein a peer with a public IP address can receive upload service from any peer in the BT system, while a peer behind firewall can only receive upload service by peers with public IP addresses.

In our model, we assume there are two classes of peers: peers with publicly routable IP address, and peers behind firewall. Let λ_p be the average rate at which non-firewalled peers arrive, and λ_f be the average rate at which firewalled peers arrive. Denote the number of non-firewalled leechers and seeders as X_p and Y_p, the number of firewalled leechers and seeders as X_f and Y_f. For simplicity of presentation, we do not differentiate peers by the amount of chunks they have cached. Similar to the previous mathematical development, we have the following differential equations to describe the dynamic of the overall system:

$$\frac{dX_p}{dt} = \lambda_p - \frac{X_p \cdot min\{\mu\rho(X_p + Y_p + X_f + Y_f), B/2\}}{sM},$$

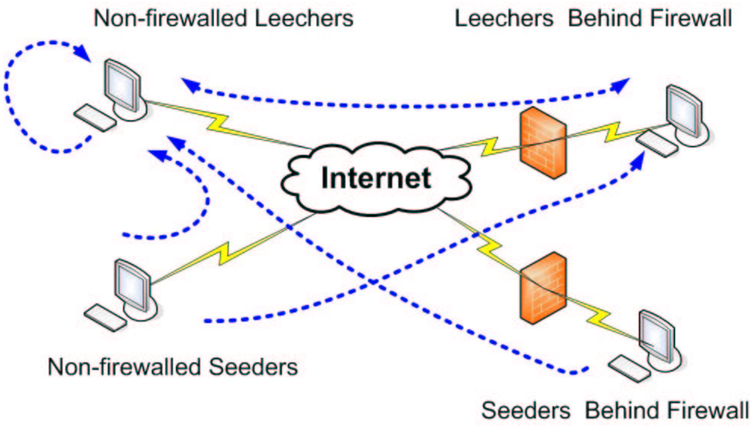


Figure 7: General model illustrates the impact of firewalls and NATs

$$\begin{aligned}
\frac{dY_p}{dt} &= \frac{X_p \cdot min\{\mu\rho(X_p + Y_p + X_f + Y_f), B/2\}}{sM} - \gamma_p Y_p, \\
\frac{dX_f}{dt} &= \lambda_f - \frac{X_f \cdot min\{\mu\rho(X_p + Y_p), B/2\}}{sM}, \\
\frac{dY_f}{dt} &= \frac{X_f \cdot min\{\mu\rho(X_p + Y_p), B/2\}}{sM} - \gamma_f Y_f. \qquad (13)
\end{aligned}$$

For mathematical tractability, we assume the situation that a peer will leave the system as soon as it obtains all the necessary chunks. This implies $Y_p(t) = 0$ and $Y_f(t) = 0$ for large t. Eq. (13) can be reduced to:

$$\begin{aligned}
\frac{dX_p}{dt} &= \lambda_p - \frac{X_p \cdot min\{\mu\rho(X_p + X_f), B/2\}}{sM}, \\
\frac{dX_f}{dt} &= \lambda_f - \frac{X_f \cdot min\{\mu\rho X_p, B/2\}}{sM}. \qquad (14)
\end{aligned}$$

We are interested in the steady state behavior and we have the following important theorems:

Theorem 5 (Equilibrium point) *When $\lambda_p > \lambda_f$, in the regime that $X_p \geq 0$ and $X_f \geq 0$, Eq. (14) has the unique equilibrium point $\bar{\mathbf{X}} = (\bar{X}_p, \bar{X}_f)$:*

$$\bar{\mathbf{X}} = \begin{cases} (\sqrt{\frac{sM(\lambda_p - \lambda_f)}{\mu\rho}}, \frac{\sqrt{sM}\lambda_f}{\sqrt{\mu\rho(\lambda_p - \lambda_f)}}) & \textit{when } \frac{2\lambda_p\sqrt{\mu\rho sM}}{\sqrt{\lambda_p - \lambda_f}} < B, \\ (\frac{2sM\lambda_p}{B}, \frac{B\lambda_f}{2\lambda_p\mu\rho}) & \textit{when } 2\sqrt{\lambda_p\mu\rho sM} < B < \frac{2\lambda_p\sqrt{\mu\rho sM}}{\sqrt{\lambda_p - \lambda_f}}, \\ (\frac{2sM\lambda_p}{B}, \frac{2sM\lambda_f}{B}) & \textit{when } 0 < B < 2\sqrt{\lambda_p\mu\rho sM}. \end{cases} \qquad (15)$$

When $\lambda_p \leq \lambda_f$, in the regime that $X_p \geq 0$ and $X_f \geq 0$, Eq. (14) has the unique equilibrium point $\bar{\mathbf{X}} = (\bar{X}_p, \bar{X}_f)$:

$$\bar{\mathbf{X}} = \begin{cases} (\frac{2sM\lambda_p}{B}, \frac{B\lambda_f}{2\lambda_p\mu\rho}) & \textit{when } 2\sqrt{\lambda_p\mu\rho sM} < B, \\ (\frac{2sM\lambda_p}{B}, \frac{2sM\lambda_f}{B}) & \textit{when } 0 < B < 2\sqrt{\lambda_p\mu\rho sM}. \end{cases} \qquad (16)$$

Proof: Due to the lack of space, please refer to the technical report [10]. ■

Theorem 6 (Local Stability) *The equilibrium point given by Theorem 5 is asymptotically stable.*

Proof: Please refer to the technical report [10]. ■

Theorem 7 *Let $\bar{T}_{d,p}$ and $\bar{T}_{d,f}$ denote the average downloading time for non-firewalled peers and peers behind firewall respectively. The average downloading times are given by:*
When $\lambda_p > \lambda_f$:

$$(\bar{T}_{d,p}, \bar{T}_{d,f}) \;=\; \begin{cases} \left(\frac{\sqrt{sM(\lambda_p-\lambda_f)}}{\sqrt{\mu\rho}\lambda_p}, \frac{\sqrt{sM}}{\sqrt{\mu\rho(\lambda_p-\lambda_f)}}\right) & \textit{when } \frac{2\lambda_p\sqrt{\mu\rho sM}}{\sqrt{\lambda_p-\lambda_f}} < B, \\ \left(\frac{2sM}{B}, \frac{B}{2\lambda_p\mu\rho}\right) & \textit{when } 2\sqrt{\lambda_p\mu\rho sM} < B < \frac{2\lambda_p\sqrt{\mu\rho sM}}{\sqrt{\lambda_p-\lambda_f}}, \\ \left(\frac{2sM}{B}, \frac{2sM}{B}\right) & \textit{when } 0 < B < 2\sqrt{\lambda_p\mu\rho sM}. \end{cases} \tag{17}$$

When $\lambda_p \leq \lambda_f$:

$$(\bar{T}_{d,p}, \bar{T}_{d,f}) \;=\; \begin{cases} \left(\frac{2sM}{B}, \frac{B}{2\lambda_p\mu\rho}\right) & \textit{when } 2\sqrt{\lambda_p\mu\rho sM} < B, \\ \left(\frac{2sM}{B}, \frac{2sM}{B}\right) & \textit{when } 0 < B < 2\sqrt{\lambda_p\mu\rho sM}. \end{cases} \tag{18}$$

Proof: By Little's result [14], $\bar{T}_{d,p}$ is given by $\bar{T}_{d,p} = \frac{\bar{X}_p}{\lambda_p}$, and $\bar{T}_{d,f}$ is given by $\bar{T}_{d,f} = \frac{\bar{X}_f}{\lambda_f}$. Based on Theorem 5, the above results can be easily derived. ■

Remark 1: Importance of non-firewalled peers: Consider the extreme case of small birth of non-firewalled peers (i.e., $\lambda_p \to 0$), under this case we have $\lambda_f > \lambda_p$ and $2\sqrt{\lambda_p\mu\rho sM} < B$. The average downloading time for non-firewalled peers is $\lim_{\lambda_p \to 0} \bar{T}_{d,p} = \frac{2sM}{B}$, which is a constant, but $\lim_{\lambda_p \to 0} \bar{T}_{d,f} = \frac{B}{2\lambda_p\mu\rho} \to \infty$, which means the peers behind firewall cannot finish the file downloading without the help of non-firewalled peers. In summary, we need to have a sufficient number of non-firewalled peers to sustain the file sharing process.

Remark 2: Performance gap: It is easy to prove that in all situations listed above, $\bar{T}_{d,p} \leq \bar{T}_{d,f}$, which implies that non-firewalled peers can always perform *at least as good as* peers behind firewalls. We define $\mathcal{G}$ as the performance gap of the downloading time between non-firewalled peer and firewalled peer. We have $\mathcal{G} = 1 - \bar{T}_{d,p}/\bar{T}_{d,f}$. When $\mathcal{G} = 0$, it means both classes of peers have the same downloading time while $\mathcal{G} = 1$ means that the firewalled peers take a very long time to complete the file download. We have the following important observations:

- When $0 < B < 2\sqrt{\lambda_p \mu \rho s M}$, which represents the situation that bandwidth of all peers are constrained, then $\mathcal{G} = 0$. This implies that the impact of firewalls is neglectable.

- When $2\sqrt{\lambda_p \mu \rho s M} < B$ (i.e., bandwidth is unconstrained), we have $\mathcal{G} < 1$ but $\mathcal{G}$ is increasing as we reduce λ_p. In other words, when there are few number of non-firewalled peers, there is a noticeable performance gap between these two classes.

- When $\lambda_p > \lambda_f$ and $\frac{2\lambda_p \sqrt{\mu \rho s M}}{\sqrt{\lambda_p - \lambda_f}} < B$, we have $\mathcal{G} = \lambda_f / \lambda_p < 1$. This implies that there is a performance gap and this gap depends on the relative arrival rates (or population) of these two classes of peers.

5 File Availability and the Chunk Selection Policies

In this section, we look at another important performance measure - the file availability for a BT-like system. A file is available only when a peer can download all the chunks needed from seeders or other peers in the system. If there is always at least one seeder in the system, naturally the file is always available. However in reality, the seeders may want to minimize the time of staying in the system and the leechers may choose to depart from the system once they obtain all necessary chunks, or they may abort in the middle of the file download due to the system or network failures. Thus the system may lose some chunks due to the departure of the peers and seeders and the remaining downloading processes will never finish. There are many factors that may influence the file availability. In this paper we are interested in how the *chunk selection algorithm* can affect the file availability. In other words, if a peer needs to download a chunk from a neighboring peer, which chunk is the proper one so as to improve the probability to complete the file download process?

5.1 Modeling the File Availability

In this section, we present a mathematical model to evaluate the file availability of a BT-like file sharing system. We still use the similar notations as in previous sections. Assume that at time t, there are n peers in the system and the intended file $\mathcal{F}$ has M chunks: $\mathcal{F}_1$, $\mathcal{F}_2$,..., $\mathcal{F}_M$. Let h_i denote the number of peers which have cached the i^{th} chunk $\mathcal{F}_i$, then h_i/n is the probability that a randomly chosen peer has this chunk $\mathcal{F}_i$. Since ρ is the *connection probability*, a peer connects to $\rho(n-1)$ number of peers on the average. Let γ_i be the probability that a peer can find $\mathcal{F}_i$ from at least one of its connecting peers, we have:

$$\gamma_i \;=\; 1 - \left(1 - \frac{h_i}{n}\right)^{\rho(n-1)} \approx \; 1 - e^{-\rho h_i}. \qquad (19)$$

Above approximation is valid for large value of n, which is usually the case for a popular BT file.

To completely download the file $\mathcal{F}$, a peer needs to collect all the M chunks. Let Θ be the probability that a peer can obtain these M chunks from its connecting peers, we have:

$$\Theta = \text{Prob[A peer can get all } M \text{ chunks]} = \prod_{i=1}^{M} \text{Prob[A peer can get } \mathcal{F}_i] = \prod_{i=1}^{M} \gamma_i = \prod_{i=1}^{M} (1 - e^{-\rho h_i}). \quad (20)$$

To gain the understanding about the appropriate chunk selection policy, we first find the optimal distribution of different types of chunks in the system. Assume that C is the total storage space (in units of chunks) of all n peers in the system, we formulate a constrained optimization problem:

$$\begin{aligned} \max \qquad & \Theta = \prod_{i=1}^{M} (1 - e^{-\rho h_i}) \\ \text{s.t.} \qquad & \sum_{i=1}^{M} h_i \leq C \;\; ; \;\; h_i \geq 0, \;\; \text{for } i \in \{1, \ldots, n\}. \end{aligned}$$

The optimal solution for the distribution of chunks is:

$$\boldsymbol{h}^* = [h_1^*, \ldots, h_M^*] = \left[\frac{C}{M}, \ldots, \frac{C}{M} \right]. \quad (21)$$

The physical meaning of the above result is not surprising: to maximize the probability of obtaining a file, the system should ensure that the chunks are as *evenly distributed* as possible across the system. We can use the following function to measure how evenly the chunks are distributed:

$$V(h_1, h_2, \ldots, h_M) \;\; = \;\; \sum_{i=1}^{M} \frac{(h_i - \bar{h})^2}{M}. \quad (22)$$

where $\bar{h} = \sum_{i=1}^{M} h_i / M$ is the average number of chunks in the system at time t. In essence, V measures the *variance* of the chunk distribution in the system. V is minimized, when $h_1 = \ldots = h_M = \bar{h}$.

Now the question we need to answer is: given the existing distribution $\boldsymbol{h} = [h_1, \ldots, h_M]$, what is the proper chunk selection policy? This can be formulated as an problem to *minimize* V because when V is close to zero, it means all chunks are evenly distributed across the system (Here the decision variables are Δh_i, $i = 1, \ldots, M$, Δh_i is the rate of change of number of $\mathcal{F}_i$).

To solve the above optimization problem, let us consider in a short period of time Δt. For $\Delta h_i \geq 0$, it is the number of newly replicated $\mathcal{F}_i$ in Δt. Assume the system is in steady-state so that the throughput of system $\bar{T}_p$ could be considered as a constant. The increase of total number of chunks copies $\sum_{i=1}^{M} \Delta h_i$ is upper bounded by $\bar{T}_p \cdot \Delta t$. To minimize $V(h_1, h_2, \ldots, h_M)$ within the range of change of $\sum_{i=1}^{M} \Delta h_i \leq \bar{T}_p \cdot \Delta t$, one can use the steepest descent method for ℓ_1-norm (see [5] page 478), we have the solution:

$$\Delta h_i \;\; = \;\; \begin{cases} \bar{T}_p \cdot \Delta t & \text{if } -\frac{\partial V}{\partial h_i} \text{ is greatest,} \\ 0 & \text{otherwise.} \end{cases} \quad (23)$$

Since $-\frac{\partial V}{\partial h_i} = \frac{2(\bar{h}-h_i)}{M}$, Eq. (23) reveals that to maximize the system measure of file availability, system should let peers download the *rarest* chunk in the system, which is indeed the chunk selection algorithm used in the BT protocol.

Mathematically, a peer should always download the rarest chunk (assuming that peer does not possess this chunk) from its neighboring peers. Practically as we will show by simulation in the later section, this policy works well when the connection probability ρ is small(i.e., peers have few neighbors). However when ρ is large(i.e., the peers are quite well connected), it may cause some problem and reduce in file availability(we will show it by simulations later). In this case, assume that $\mathcal{F}_i$ is the rarest chunk and $\mathcal{F}_j$ is the second to the rarest chunk in the system. Due to the large connection probability ρ, nearly all peers prefer to download $\mathcal{F}_i$ and those peers that hold on to $\mathcal{F}_j$ depart or abort from the system, then the file will *not* be available. This synchronization problem deteriorates the availability especially among the system with high connectivity where peers may have many neighbors.

To alleviate this problem, we propose the *file availability enhancement* (FAE) algorithm. In essence, it tries to *randomize* the chunk selection process but the *rarest* chunk will still be selected with the *highest* probability. We define Δh_i as:

$$\Delta h_i = \begin{cases} \frac{\partial V}{\partial h_i} = \frac{2(\bar{h}-h_i)}{M} & \text{if } h_i \leq \bar{h} \\ 0 & \text{otherwise.} \end{cases}$$

Among all its missing chunks, a peer will select $\mathcal{F}_i$ with the probability σ_i where

$$\sigma_i = \frac{\Delta h_i}{\sum_{\forall \Delta h_j > 0} \Delta h_j}. \tag{24}$$

Note that for the above discussion, the value of h_i is obtained by examining all n peers in the system, which implies peers know the *global* information. In a practical implementation, a peer can only connect to a subset of peers. In this case, the value of h_i is the number of $\mathcal{F}_i$ from its neighbors, which is just the *local* information. In the following we consider algorithms in both cases: with global information or with local information. Now we have the following chunk selection algorithms:

- **Global Rarest First (GRF)**: A peer will select $\mathcal{F}_i$ from a neighboring peer with probability 1, where $\mathcal{F}_i$ is the rarest chunk in the whole system.

- **Local Rarest First (LRF)**: A peer will select $\mathcal{F}_i$ from a neighboring peer with probability 1, where $\mathcal{F}_i$ is the rarest chunk among its connecting peers. This is the built-in chunk selection algorithm in BitTorrent system.

- **Global File Availability Enhancement (GFAE)**: A peer will select $\mathcal{F}_i$ from a neighboring peer with probability σ_i, which is calculated by the global information h_i for $i = 1 \ldots M$.

- **Local File Availability Enhancement (LFAE)**: A peer will select $\mathcal{F}_i$ from a neighboring peer with probability σ_i, which is calculated by the local information h_i for $i = 1 \ldots M$.

- **Random Selection (RD)**: A peer will select $\mathcal{F}_i$ from a neighboring peer assuming $\mathcal{F}_i$ is one of its missing chunk which is cached by the neighboring peer.

Note that, GRF and GFAE require global information for peers to make their decisions, which can hardly be implemented in real system. So we just use the results of these two policies as benchmarks.

5.2 Performance of Different Chunk Selection Algorithms

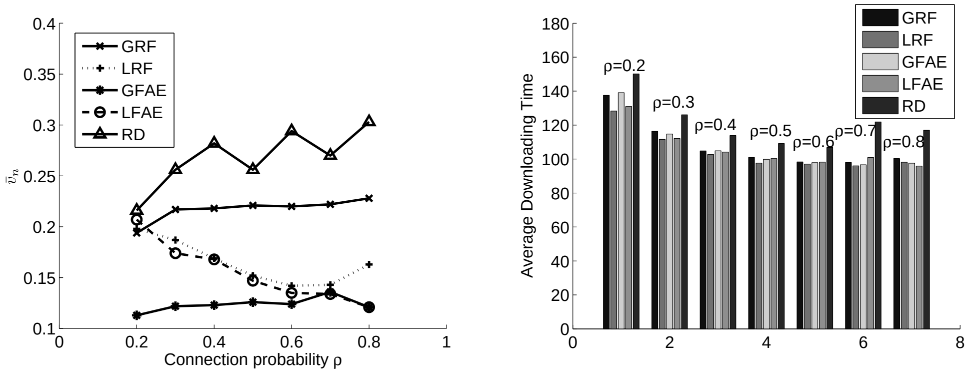
In this section, we carry out simulations to compare the effect on average downloading time and file availability for different chunk selection algorithms described in previous subsection. In each of the simulation, we allow peers to dynamically join or leave the system. The arrival process of peer is a Poisson process. A peer can leave the system after obtaining all the necessary chunks, or may abort in the middle of the file download. In each experiment, the served file has 200 chunks. An initial seeder is put in the system and this seeder stays in the system from $t = 0$ to $t = 500$. All other peers may abort the system before collecting all chunks at the abortion rate θ, and choose the seeding time according to the leaving rate γ after they become seeders.

Note that we use the variance measure V defined in Eq. (22) to measure the goodness of the chunk selection algorithm. Since V depends heavily on the number of peers, while in our simulation, the number of peers are time varying (due to peer's arrival and departure). So we define a normalized metric:

$$v_n(t) = \frac{\sqrt{V(t)}}{\bar{h}(t)},$$

which is used to measure the variance normalized by the average number of chunks at time t. We use the mean $\bar{v}_n$ of observed $v_n(t)$ from time 400 to time 1500.

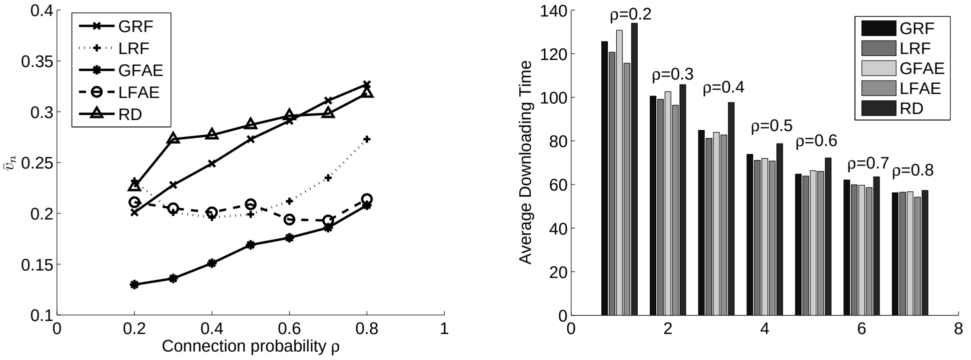
Experiment 1: Normalized Variance and File Downloading Time under Low Bandwidth Scenario: In this experiment, we fix the bandwidth for each peer to be $B = 4.5$, arrival rate $\lambda = 0.4$, leaving rate $\gamma = 0.6$, abortion rate $\theta = 0.01$ and transfer rate $\mu = 0.3$. We vary the connectivity probability ρ from 0.2 to 0.8. Fig. 8(a) illustrates the normalized variance for the five chunk selection algorithms. Note that GFAE and LFAE provide better availability and the random policy is the worst. It is interesting to note that LRF even performs better than GRF especially when ρ is high, although LRF only uses the local information. From the trace file of our simulation we find the justification that when ρ is high, peers get information from most of the peers in the system. So the GRF is more likely to cause the synchronization problem, which means all peers tends to download the few chunks that are the rarest. LRF brings more randomness to alleviate this problem. Our FAE with local or global information is better than LRF when ρ is high because we make a probabilistic choice to remedy this problem. Another important observation is that when we increase ρ, the availability is also improved by LEF and LFAE. This is because in this simulation setting we set bandwidth to $B = 4.5$, so peers can not perform more downloading due to the bandwidth constraint. Even when we increase ρ so that peers may have more neighbors, they can still download from a small part of all its the neighbors. This randomness pushes system away from this synchronization problem.



(a) Availability, $B = 4.5$

(b) Average downloading time, $B = 4.5$

Figure 8: Availability and throughput by different chunk selection polices in low bandwidth case.



(a) Availability, $B = 12$

(b) Average downloading time, $B = 12$

Figure 9: Availability and throughput by different chunk selection polices in high bandwidth case.

In terms of average downloading time, from Fig. 8(b) we find that the performance of different policies are actually comparable except the Random policy. Random policy performs worst because it can not distribute all types of chunks evenly among peers so peers may suffer due to waiting for useful chunks. The important point is that the GFAE and LFAE provide similar average downloading time as compared with GRF and LRF, yet, GFAE and LFAE have better availability.

Experiment 2: Normalized Variance and File Downloading Time under High Bandwidth Scenario: In this simulation, we set bandwidth $B = 12$ so that we simulate the case that peers have high bandwidth connection to download the file. In this setting, GFAE is the best in terms of the normalized variance. LFAE performs better than LRF especially when ρ is high and LRF performs better than GRF. Random policy is still the worst among the all. We observe that the availability deteriorates when ρ increases. This is due to the fact that increasing ρ may introduce the synchronization problem, but LFAE is less sensitive in this

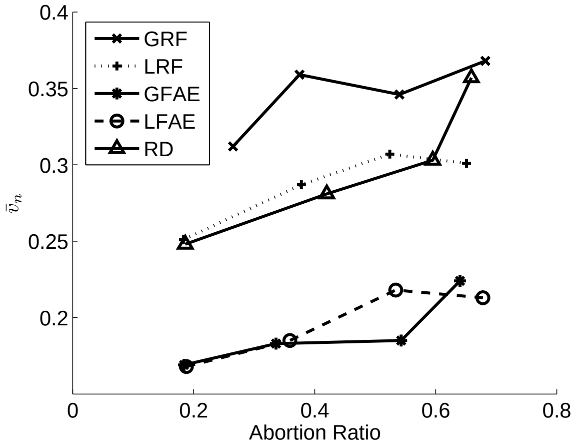


Figure 10: Availability by different chunk selection polices in severely dynamic system.

regard.

For average downloading time, random policy is still much worse than the others when ρ is small. Random policy in this situation can not ensure the chunks equally distributed across the system because peers have only few choice due to the small number of neighbors. But when ρ is large, Random policy has similar performance as compared with the others.

Experiment 3: Normalized Variance under Different Peer's Abortion Rates:

In this experiment, we increase arrival rate $\lambda = 0.6$, and vary different abortion rate θ from 0.005 to 0.02 to investigate the performance of the system with high arrivals and abortion. In Fig. 10, the X-axis represents the fraction of peers that abort before downloading all the chunks in the system. From this figure we can observe that the GFAE or LFAE has a lower value of the normalized variance, this implies high file availability at these extreme conditions even when $\theta = 0.02$ and nearly 70% peers abort before obtaining all chunks.

6 Conclusion

In this paper, we first propose a fluid model based on the stochastic differential equation method in modeling and characterizing the peer behaviors and performance metrics of BT-like P2P systems. We obtain the closed-form solution of the average number of seeders and leechers, as well as the average file downloading time and the steady state system throughput. We validate this model by the discrete event simulator, and find our model has much higher accuracy, while previous model proposed in [18] may provide wrong performance estimates under large system settings. Based on the closed-form solution, we quantify the sensitivity of the downloading time to various system parameters such as peers' arrival rate, seeder's departure rate, connection probability and transmission bandwidth. We also extend the model to investigate the impact of firewalls or NATs on the performance of BT-like system. We find that peers in the public domain play an important role and analyze the performance gap between these two classes of peers. Lastly, we investigate the file availability issue in terms of chunk selection algorithms. We model the file availability and find that the rarest first is the theoretical solution to maximize the availability. In practice, however, one may encounter the synchronization problem in using the rarest first policy especially in high connectivity scenario. To alleviate this problem we propose a *randomized version* of the chunk selection policy. We show the experimental results of all these algorithms and illustrate our proposed algorithm can significantly improve the file availability of BT systems.

References

[1] Bittorrent protocol. http://www.bittorrent.com/protocol.html.

[2] D. Arthur and R. Panigraphy. Analyzing the efficiency of bittorrent and related peer-to-peer networks. In *SODA*, Januray 2005.

[3] A. Bellissimo, P. Shenoy, and B. N. Levine. Exploring the use of bittorrent as the basis for a large trace repository. Technical report, June 2004.

[4] A. Bharambe, C. Herley, and V. Padmanabhan. Understanding and deconstructing bittorrent performance. In *Proc. ACM SIGMETRICS*, 2005.

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

[6] R. Brockett. Stochastic control. Lecure notes.

[7] F. Clvenot-Perronnin and K. R. P. Nain. Multiclass p2p networks: Static resource allocation for service differentiation and bandwidth diversity. In *Performance*, 2005.

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

[9] G. de Vecianna and X. Yang. Fairness, incentives and performance in peer-to-peer networks. In *the Forty-first ANnual Allerton Conference on Communication, Control and Computing*, 2003.

[10] B. Fan, D.-M. Chiu, and J. C. S. Lui. On the performance and availability of bittorrent-like p2p file sharing systems. http://www.cse.cuhk.edu.hk/˜ bfan/TechReport20051.pdf, Nov 2005.

[11] C. Gkantsidis and P. Rodriguez. Network coding for large scale content distribution. In *Proc. IEEE INFOCOM*, 2005.

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

[13] 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*, Apr 2004.

[14] L. Kleinrock. *Queueing Systems*. Wiley-Interscience, 1976.

[15] T. B. Ma, S. C. M. Lee, J. C. S. Lui, and D. K. Y. Yau. A game theoretic approach to provide incentive and service differentiation in p2p networks. In *ACM SIGMETRICS/PERFORMANCE*, June 2004.

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

[17] 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*, Feb 2005.

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

[19] S. M. Ross. *Stochastic Processes*. John Wiley, New York, 1983.

[20] K. Skevik, V. Goebel, and T. Plagemann. Analysis of bittorrent and its use for the design of a p2p based streaming protocol for a hybrid cdn. Technical report, June 2004.

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

[22] X. Zhang, J. Liu, B. Li, and T. Yum. Coolstreaming/donet: A data-driven overlay network for live media streaming. In *IEEE INFOCOM*, Miami, FL, USA, March 2005.