

Extended Equal Service and Differentiated Service Models for Peer-to-Peer File Sharing

Jianwei Zhang¹, Yongchao Wang², Wei Xing^{1,*}, Dongming Lu¹

¹*College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China*

²*Library and Information Center, Zhejiang University, Hangzhou 310027, China*

Abstract

Peer-to-Peer (P2P) systems have proved to be the most effective and popular file sharing applications in recent years. Previous studies mainly focus on the equal service and the differentiated service strategies when peers have no initial data before their download. In an upload-constrained P2P file sharing system, we model both the equal service process and the differentiated service process when peers' initial data distribution satisfies some special conditions, and also show how to minimize the time to get the file to any number of peers. The proposed models can reveal the intrinsic relations among the initial data amount, the size of peer set and the minimum last finish time. By using the models, we can also provide arbitrary degree of differentiated service to a certain number of peers. We believe that our analysis process and achieved theoretical results could provide fundamental insights into studies on bandwidth allocation and data scheduling, and can give helpful reference both for improving system performance and building effective incentive mechanism in P2P file sharing systems.

Keywords: Peer-to-Peer; file sharing; equal service; differentiated service

1 Introduction

Peer-to-Peer (P2P) systems have proved to be the most effective and popular file sharing application in recent years. BitTorrent (BT) [1] is the most popular P2P application due to its high degree of scalability and effective incentive mechanism to reduce free-riding.

Recently, most of studies on P2P systems are focused on social network, incentive mechanism [2][3], piece selection algorithm [4] (especially directed at media streaming) etc. However, the bandwidth allocation and data scheduling for differentiated service in application layer has been paid little attention. We consider this problem to be one of the most fundamental and important problems because it determines how the data and bandwidth resource is allocated among peers and plays a decisive role in respect of improving the performance of P2P systems. In addition, theoretical analysis for differentiated service can also give some insight into the design of incentive mechanism.

*Corresponding author.

Email address: wxing@zju.edu.cn (Wei Xing).

There have been a lot of analytical studies focused on the overlay topology, service capacity and piece scheduling etc., when peers are homogeneous with the same download and upload bandwidth. Yang et al. [5][6] propose a branching process for transient state and a Markov chain model for steady state of P2P systems. Qiu et al. [7] present a simple fluid model in BitTorrent-like system for the first time, and they evaluate the scalability, performance and efficiency of such kind of P2P systems. However, this model only focuses on the peer arrival rate, departure rate in steady state, and makes an unrealistic assumption that all peers have homogeneous upload bandwidth. Based on the fluid model in [7], Clevenot-Perronnin et al. [8] propose a two-set fluid model for service differentiation in BitTorrent-like content distribution systems, whereas they assume peers in the same class have the same bandwidth. Biersack et al. [9] propose three typical analytical models, i.e. linear model, k -tree model, k -Ptree model, to address the problem that how long it should take to distribute a file to N peers. Results indicate that the service capacity of most P2P systems grows exponentially with the number of pieces in a file. Esposito et al. [10] adapt the fluid model in [7] to transient phase of P2P file disseminating and propose a seed scheduling algorithm called *proportional fair seed scheduling* (PFS). They show that their solution can effectively reduce the average download time than the standard BitTorrent protocol.

Under a more general condition that peers have heterogeneous bandwidth, some studies investigate the system performance (mostly in terms of last finish time or average finish time) and fairness (mostly in terms of peers' contributions to the system) to provide fundamental principles and criteria for P2P protocol and application system design. Wu et al. [11] propose a *centrally scheduled file distribution* (CSFD) protocol that can provably minimize the last finish time of a single source file distribution process. Mundinger et al. [12][13] characterize the problem that how to minimize the last finish time when peers have heterogeneous upload bandwidth and the file can only be divided into discrete pieces. They also evaluate the minimum last finish time when all peers receive equal service and the file can be divided into infinitely small pieces, and show how each peer should schedule its bandwidth and data to achieve the lower bound. Kumar et al. [14] investigate file distribution with differentiated service in two sets of peers, with the purpose of distributing the file as quickly as possible to all the first-set peers. Also, they derive explicit expressions for the minimum distribution time. They make an assumption that the second-set peers have no initial data before download. Mehryar et al. [15] extend the results in [14] and examine different performance criteria (e.g. last finish time, average finish time, etc.) by considering how the pieces and the bandwidth resources can be allocated in order to reach the theoretical bounds. Fan et al. [16] analyze the delicate tradeoff between system performance and system fairness in BitTorrent-like system, and show that the default configuration in BitTorrent puts more emphasis on fairness rather than performance. Chan et al. [17] deduce a theoretical bound on the minimum last finish time in terms of discrete cycles. They also evaluate several scheduling algorithm suitable for the push-based model, e.g., *Rarest piece first*, *Most Demanding node first* and the *Maximum-flow* algorithm, and simulation results show that the *Maximum-flow* algorithm outperforms all other algorithms. Ezovski et al. [18] show how each peer should use its upload bandwidth to minimize the average finish time in a fully-connected network without the presence of peer leaving by using the *water-filling technique*. Wen et al. [19] extend Ezovski et al.'s studies by taking peer leaving condition into account. They present the *reversed water filling algorithm* (RWF) in order to minimize the average finish time. However, no theoretical lower bound is calculated because of the complexity of the problem.

This paper is inspired by the seminal studies [12][14][15][18][19], and can be regarded as an extension of [14][15][18]. In addition, the analysis methods (e.g. the *water-filling technique*) in

[18][19] have some significant implications for our research. To the best of our knowledge, there is no theoretical analysis towards the file sharing in P2P network where peers with heterogeneous upload bandwidth already have some initial data before their download. In [12][15], the equal service model and the two-set differentiated service model (including the *Multiplicity Theorem*) are first proposed, and the system performance is also evaluated in terms of the last finish time, etc. In [14], towards the multi-set problem, they consider a two-set system for simplicity, nevertheless, how the second-set peers can finish their download has not been further considered. Based on their work, we present extended equal service model and differentiated service model when peers' initial data distribution satisfies some special conditions (*UCP condition*), deduce the φ -*Multiplicity Theorem*, and show how to make the best use of all peers' upload bandwidth to minimize the last finish time for peers in the superior set. Both our analysis process and achieved theoretical results could provide fundamental insights into studies on bandwidth allocation and data scheduling, and can give helpful reference both for improving system performance and building effective incentive mechanism in P2P file sharing systems.

The rest of this paper is organized as follows. Section 2 sums up and refines a few fundamental results in previous work. Section 3 discusses the concept of multiplicity in detail and presents our extended equal service and differentiated service models. Some numerical results are given in section 4 and we make the conclusion in section 5.

2 Preliminary

2.1 Assumptions and Metrics

According to the actual application background and the need of the analysis, most of related works make the following reasonable assumptions, which will also be adopted in this paper:

- The file can be divided into infinitesimal pieces. Each peer can simultaneously upload to as many peers as needed. Although in real P2P application systems, the maximum concurrent uploads is a fixed value. This assumption is consistent with [12][13][14][18] etc.
- Upload bandwidth rather than download bandwidth is the only bottleneck. This assumption holds because in reality users always have asymmetric access bandwidth. This assumption is consistent with [14][15][18][19] etc.

Unlike the P2P live streaming case where emphasis should be given onto the transmission delay [20] and the streaming capacity [21] etc., the following two metrics involving the download time should be considered in detail, as we assume that a file can only be used after its last bit is obtained by a peer.

- *Last finish time*: The time interval every peer in a specified peer set obtains the whole file.
- *Bottleneck time*: The time interval that the source can distribute its fresh data (the data no one else has) to the swarm.

Obviously, for any set of peers, the last finish time is no less than the bottleneck time. One of the most important targets in this paper is to minimize the last finish time.

2.2 Fundamental Results in Previous Work

Given an upload-constrained fully connected P2P network, if all of N peers receive the same service (*equal service*) from a source, the minimum last finish time T_L is given [12][14] by the following equation with symbols modified for consistency, where C_0 is the source's upload bandwidth, U is the sum of all peers' upload bandwidth, and F is the file size.

$$T_L = \frac{F}{\min \left\{ C_0, \frac{C_0 + U}{N} \right\}} \quad (1)$$

If there are two sets of peers, and the first L peers have superior priority (*differentiated service*), the minimum last finish time of peers in the first set is given [14] by the following equation, where C_{N-L} is the total upload bandwidth of the last $N - L$ peers.

$$T_L = \frac{F}{\min \left\{ C_0, \frac{1}{L} \left(C_0 + U - \frac{C_{N-L}}{L} \right) \right\}} \quad (2)$$

The bottleneck time t_b is calculated by its definition as:

$$t_b = \frac{F}{C_0} \quad (3)$$

Furthermore, Mehyar et al. [15] propose the following theorem to calculate at most how many peers can finish their download in the bottleneck time.

Multiplicity Theorem: It is possible to let the first M peers finish their download in the bottleneck time t_b if and only if:

$$C_0 \leq \sum_{i=1}^M \frac{c_i}{M-1} + \sum_{i=M+1}^N \frac{c_i}{M}, \quad (4)$$

where c_i is the upload bandwidth of peer i . For convenience of expression, we use C_M and C_{N-M} to denote the total upload bandwidth of the first L peers and the last $N - L$ peers respectively, hence above inequality can be rewritten as $C_0 \leq \frac{1}{M-1}C_M + \frac{1}{M}C_{N-M}$.

For a given P2P system, *multiplicity* is defined as the largest M such that inequality (4) holds.

It is worth noting that, the *multiplicity* here refers specifically to the M in the case that all peers are put in descending order by their upload bandwidth, as M can reach the maximum in this case. In this paper, the *multiplicity* we refer to is the largest integer M such that inequality (4) holds, where peers are put in random order in advance. The discussions in what follows will not put a limit on the order of peers either.

Owing to space constraints, the strategies to achieve the above theoretical lower (upper) bounds in related literatures will not be given in this paper.

2.3 Differentiated service model when peers have no initial data

As section 2.2 shows, T_L has already been deduced, however, there is no unified form for T_L in the differentiated service model [14] and the *Multiplicity Theorem* [15]. In this section, we will

reformulate T_L when there is no initial data before download.

For convenience of expression, we define the *multiplicity function*:

$$F(M) = \frac{1}{M-1}C_M + \frac{1}{M}C_{N-M} \quad (5)$$

When $M = N$, $F(M)$ is defined by $\frac{1}{M-1}C_M$; when $M = 1$, $F(M)$ becomes infinity [15].

According to the magnitude of L , there are two cases to be considered.

1) $L \leq M$

In this case, the last finish time of the first L peers is equivalent to the bottleneck time, i.e.:

$$T_L = t_b \quad (6)$$

2) $L > M$

Since $L > M$, we can deduce $F(L) < F(M)$. In this case, the first L peers can reach the maximum download rate, and T_L can be calculated as follows:

$$T_L = \frac{F}{\frac{1}{L}(C_0 + U - \frac{1}{L}C_{N-L})} \quad (7)$$

3 Extended equal service and differentiated service models

3.1 Data Distribution

Researches in [12][13][14][15] etc. all make an assumption that each peer has no data before download. In this paper, in order to study how the initial data will put an effect on the whole file dissemination process, we will consider the cases when the data distribution satisfies two special conditions.

According to the previous section, in order to provide peers in $L1$ with superior service, peer $L+i$ in $L2$ has to download at certain rate $\mu \leq \frac{1}{L}c_{L+i}$ from the source, then replicates this data stream to all L peers in $L1$. In the case of $L > M$, when all L peers finish downloading in T_L , the data amount peer $L+i$ has is $\frac{1}{L}c_{L+i}$ (refer to Fig.1). As a consequence, any peer in $L2$ has entirely different data proportional to its upload bandwidth from other peers in $L2$ after the download process. Obviously, the initial data that peers in $L2$ already have after the current download process will directly influence how these peers can finish their download in further download process. In the remaining sections, we call these data the *forwarding data*. Now we introduce the *UP condition* as follows:

Unique and Proportional condition: (*UP condition*) The data contained within the same peer set is a subset of the whole file, and it has the following two characteristics:

- Each file piece must be owned by only one peer (*unique data*);
- The ratio of the unique data amount to the upload bandwidth of each peer is equal.

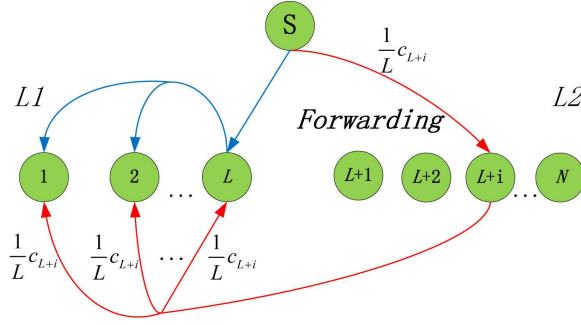


Fig. 1: Illustration of the *UP condition*

Let a_i denote the data peer i owns before download, and the ratio of all peers' data amount to the file size F is denoted by φ (when φ is 0, it turns to the no-data case), i.e.:

$$\varphi = \sum_{i=1}^N a_i / F \quad (8)$$

We can formally describe the *UP condition* as follows:

$$\left\{ \begin{array}{l} t_a := \frac{a_1}{c_1} = \frac{a_2}{c_2} = \dots = \frac{a_N}{c_N} = \frac{\varphi F}{U} \\ a_i \cap a_j = \emptyset, \forall i \neq j \\ \bigcup_{i=1}^N a_i \subseteq F \end{array} \right., \quad (9)$$

where t_a is defined as the time interval that each peer can distribute its own unique data to other peers. According to Equation (9) and the definition of φ , we have $\varphi \leq 1$.

As a supplement to the *UP condition*, we also introduce the *UCP condition* as follows:

Unique-Common and Proportional condition: (*UCP condition*) The data contained within the same peer set is a subset of the whole file, and it has the following two characteristics:

- Each file piece must be owned by only one peer (*unique data*) or be owned by all peers (*common data*);
- The ratio of the unique data amount to the upload bandwidth of each peer is equal.

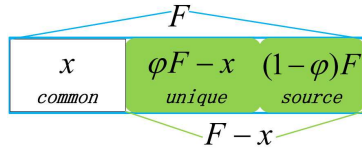


Fig. 2: Data distribution (*UP condition* and *UCP condition*)

Assume the common data, the unique data and the data owned only by the source be x , $\varphi F - x$ and $(1 - \varphi)F$ respectively, as Fig. 2 shows. The *UCP condition* for a file F is equivalent to the

UP condition for a file $F - x$, hence, without loss of generality, we only consider the *UP condition* in our extended models.

The data amount owned by all peers is denoted by F_a , and the data amount owned only by the source is denoted by F_0 , then:

$$F_a := \sum_1^N a_i = \varphi F, F_0 := (1 - \varphi)F \quad (10)$$

In order to download the whole file, the source has to upload data F_0 to other peers. Let t_0 be the bottleneck time, then:

$$t_0 = \frac{F_0}{C_0} \quad (11)$$

t_0 must be the tightly lower bound of T_L , therefore we have:

$$T_L \geq t_0 \quad (12)$$

If $t_a = t_0$, φ can be expressed as $\frac{U}{C_0 + U}$, and this value is denoted by φ_0 for convenience, as equation (13) shows. Therefore $\varphi \geq \varphi_0$ is equivalent to $t_0 \leq t_a$, and vice versa.

$$\varphi_0 := \frac{U}{C_0 + U} \quad (13)$$

3.2 Equal Service

In this section, we will extend the single-set problem mentioned in [14] when peers' initial data distribution satisfies the *UP condition*, and calculate T_L according to the relationship between φ and φ_0 , there are two cases to be considered.

1) $\varphi \geq \varphi_0$ ($t_0 \leq t_a$)

Without N peers' own upload bandwidth, it takes the source Nt_0 to send F_0 to all peers, and without the source's help, it takes all N peers $(N - 1)t_a$ to get F_a . According to the relationship of Nt_0 and $(N - 1)t_a$, we have the following two cases.

a) $(N - 1)t_a \geq Nt_0$

In this case, the source should divide part of its bandwidth x to help peers downloading F_a , thus we have (refer to Fig. 3(a)):

$$\frac{F_0}{\frac{1}{N}(C_0 - x)} = \frac{F_a}{\frac{1}{N-1}(U + x)} = T_L \quad (14)$$

When $\varphi = 1$, peers have the entire file F . The source peer can help every peer send its own data to other $N - 1$ peers, and T_L can be directly expressed as $\frac{(N-1)F}{C_0 + U}$.

b) $(N - 1)t_a < Nt_0$

In this case, we have $t_0 \leq (N-1)t_a < Nt_0$. We can find that by taking some bandwidth x out of U to download F_0 , it will take the same amount of time to download F_0 and F_a (refer to Fig. 3(b)), then we have:

$$\frac{F_0}{\frac{1}{N}(C_0 + x)} = \frac{F_a}{\frac{1}{N-1}(U - x)} = T_L \quad (15)$$

Equations (14) and (15) have the same solution:

$$T_L = \frac{(N - \varphi)F}{C_0 + U} \quad (16)$$

As all upload bandwidth can be utilized, T_L can also be viewed as the ratio of the total data required to transmit to the total upload bandwidth. Obviously, equation (16) still holds when $\varphi = 1$.

2) $\varphi < \varphi_0$ ($t_0 > t_a$)

a) $(N-1)t_a \geq t_0$

In this case, we have $Nt_0 > (N-1)t_a \geq t_0$, part of U is used to exchange F_0 , the bandwidth can also be allocated as Fig. 3(b) shows, and T_L can also be calculated by equation (16).

b) $(N-1)t_a < t_0$

Let U_N be the total upload bandwidth needed to distribute F_a to all peers, then the remaining upload bandwidth $U_{r,N}$ is $U - U_N$, i.e.:

$$U_N = \frac{(N-1)F_a}{t_0}, U_{r,N} = U - U_N \quad (17)$$

As peers must download F_0 from the source, the download speed of F_0 depends on the relation between $U_{r,N}$ and C_0 :

i) If $U_{r,N} \geq (N-1)C_0$, $U_{r,N}$ is large enough to ensure the download speed of F_0 to be C_0 . As a result, the last finish time is constraint by the bottleneck time, i.e. $T_L = t_0$.

ii) If $U_{r,N} < (N-1)C_0$, $U_{r,N}$ cannot maximize the download speed of F_0 . We can also use the bandwidth allocation strategy in equation (15) (refer to Fig. 3(b)), and T_L can also be calculated by equation (16).

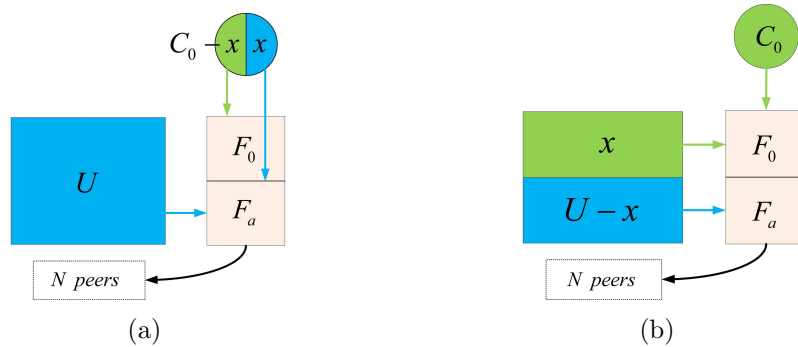


Fig. 3: Bandwidth allocation strategies (*equal service*)

3.3 Differentiated Service

In this section, the multi-set problem mentioned in [14] and the *Multiplicity Theorem* mentioned in [15] will be extended and generalized under the *UP condition*.

1) $\varphi \geq \varphi_0$ ($t_0 \leq t_a$)

If $\varphi > \varphi_0$, no one can finish its download in t_0 , i.e. $M = 0$, while in the case of $\varphi = \varphi_0$, there is one and only one can complete in t_0 , i.e. $M = 1$.

a) $L = 1$

In this case, the source and all peers from the second peer upload to the first peer at full speed, the minimum last finish time can be calculated as:

$$T_L = \frac{F - a_1}{C_0 + U - c_1} \quad (18)$$

b) $L > 1$

Denote the data amount of the first L peers and the last $N - L$ peers by F_L and F_{N-L} respectively. If C_{N-L} is fully used for transmitting F_{N-L} to the first L peers, the time interval is Lt_a , then, in order to exchange F_L in Lt_a , $\frac{L-1}{L}C_L$ of their total upload bandwidth is used and $\frac{1}{L}C_L$ is left.

Without the source's help, if all of $\frac{1}{L}C_L$ is still used to download F_a , suppose it takes the first L peers T_L' to obtain F_a , we have $t_a \leq (L-1)t_a < T_L' < Lt_a$, which means that all of $\frac{1}{L}C_L$ can be fully utilized. Suppose the first L peers divide part of their bandwidth x to help the last $N - L$ peers downloading F_{N-L} (refer to Fig. 4), then we have:

$$\frac{F_L}{\frac{1}{L-1}(C_L - x)} = \frac{F_{N-L}}{\frac{1}{L}(C_{N-L} + x)} = T_L' \quad (19)$$

Solving above equation, we have:

$$T_L' = \frac{LF_a - F_L}{U} \quad (20)$$

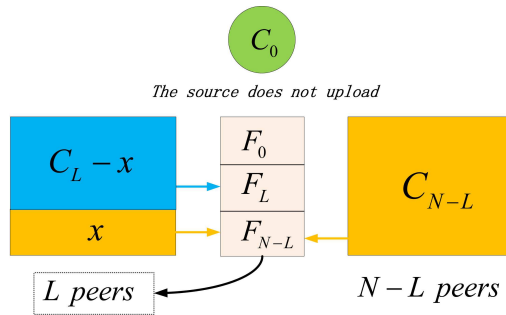


Fig. 4: Bandwidth allocation strategy without the source's help (*differentiated service*)

Without using $\frac{1}{L}C_L$, it takes the source at most Lt_0 to send F_0 to the first L peers. The most important question of the bandwidth allocation strategy is how $\frac{1}{L}C_L$ should be allocated.

If the first L peers can fully exchange their own data F_L within T , the upload bandwidth required is denoted by $U_L(L, T)$. Similarly, the upload bandwidth required to make the last $N - L$ peers upload their own data F_{N-L} to the first L peers within T is denoted by $U_{N-L}(L, T)$. Therefore we have:

$$U_L(L, T) = \frac{(L-1)F_L}{T}, U_{N-L}(L, T) = \frac{LF_{N-L}}{T} \quad (21)$$

According to the relationship between Lt_0 and T_L' , we have the following two cases.

i) $Lt_0 < T_L'$

If φ is larger than some critical value, the source should split its upload bandwidth into three parts $C_0 - x_L - x_{N-L}$, x_L and x_{N-L} to help downloading F_0 , F_L and F_{N-L} respectively in the same time T_L , as Fig. 5(a) shows. Then we have:

$$\frac{F_0}{\frac{1}{L}(C_0 - x_L - x_{N-L})} = T_L \quad (22)$$

$$(U_L(L, T_L) - x_L) + (U_{N-L}(L, T_L) - x_{N-L} - C_{N-L}) = C_L \quad (23)$$

Combining equation (22) and (23), we obtain:

$$T_L = \frac{LF - F_L}{C_0 + U} \quad (24)$$

When $\varphi = 1$, peers have the entire file F , then $t_0 = 0$. All of the source's upload bandwidth is used to transmit F_L and F_{N-L} , then we have:

$$x_L + x_{N-L} = C_0, F_L + F_{N-L} = F \quad (25)$$

Combining equation (25) and (23), we get the same solution for T_L , as equation (24) shows.

ii) $Lt_0 \geq T_L'$

In this case, part of C_L should be used to download F_0 , and we denote it by $U_{r,L,N-L}(L, T)$, then:

$$U_{r,L,N-L}(L, T) = U - U_L(L, T) - U_{N-L}(L, T) \quad (26)$$

In order to make F_L , F_{N-L} and F_0 finished in the same time T_L , the upload bandwidth can be allocated as Fig. 5(b) shows, and the following equation must hold:

$$\frac{F_0}{\frac{1}{L}[C_0 + U_{r,L,N-L}(L, T_L)]} = T_L \quad (27)$$

Solving above equation, T_L is also expressed as equation (24).

Notice that, all above bandwidth allocation strategies in i) and ii) do not need the last $N - L$ peers to forward any part of F_0 from the source.

2) $\varphi < \varphi_0$ ($t_0 > t_a$)

Suppose $L't_a < t_0 \leq (L' + 1)t_a$ ($L' \in [1, N - 1]$) or $(L' - 1)t_a < t_0$ ($L' = N - 1$), where L' is an integer. For convenience of expression, we use $L't_a < t_0 \leq (L' + 1)t_a$ ($L' \in [1, N - 1]$) to indicate the above two cases in what follows, it has no impact on the correctness of the analysis. There are three cases to be considered according to the relationship between L and $L' + 1$.

a) $L = L' + 1$

In this case, T_L' may be equal to, larger or smaller than t_0 .

i) $T_L' > t_0$

We define T_0 as the minimum downloading time of F_0 by using $\frac{1}{L}C_L$, i.e.:

$$T_0 := \max\left(\frac{F_0}{\frac{1}{L}(C_0 + \frac{1}{L}C_L)}, t_0\right) \quad (28)$$

Case A: $T_0 > Lt_a$. In this case, as can be illustrated by Fig. 5(c), part of upload bandwidth of the last $N - L$ peers should be used for forwarding F_0 from source to the first L peers, which is denoted by $U_{r,N-L}(L, T)$. Correspondingly, part of upload bandwidth of the first L peers should also be used for downloading F_0 , which is denoted by $U_{r,L}(L, T)$. Then we have:

$$U_{r,L}(L, T) = C_L - U_L(L, T), U_{r,N-L}(L, T) = C_{N-L} - U_{N-L}(L, T) \quad (29)$$

In order to make F_0 finished in T_L , we have:

$$\frac{F_0}{\frac{1}{L}[C_0 + (1 - \frac{1}{L})U_{r,N-L}(L, T_L) + U_{r,L}(L, T_L)]} = T_L \quad (30)$$

Notice that, $U_{r,N-L}(L, T_L)$ must be multiplied by a factor $1 - \frac{1}{L}$, as the source must provide data $\frac{1}{L}U_{r,N-L}(L, t_0)$ to the last $N - L$ peers. This can be illustrated by Fig. 1. Most of all, the data distribution of the last $N - L$ peers will satisfy the *UCP condition* as well after the first L peers finish their download.

Solving equation (30), we get:

$$T_L = \frac{(L - \varphi)F}{C_0 + U - \frac{1}{L}C_{N-L}} \quad (31)$$

Case B: $T_0 \leq Lt_a$. In this case, $\frac{1}{L}C_L$ can be fully used for downloading F_0 in a time interval smaller than Lt_a , then we can use the same bandwidth allocation strategy as Fig. 5(b) shows to make F_0 and F_a finished in the same time T_L , and T_L can be also expressed as equation (24).

ii) $T_L' \leq t_0$

Case A: $T_0 > Lt_a$. This case is similar to case A of i), and T_L can be calculated in the same way (refer to Fig. 4 (c)), as equation (31) shows.

Case B: $T_0 \leq Lt_a$. If $t_0 < (L' + 1)t_a$, F_L and F_{N-L} cannot be finished in t_0 without using $\frac{1}{L}C_L$. From equations (21) and (26), we can calculate $U_{r,L,N-L}(L, t_0) = U - \frac{(L-1)F + F_{N-L}}{t_0}$. In order to make the first L peers finish downloading F_0 within t_0 , we can allocate all upload bandwidth as Fig. 5(b) shows, and the following inequality must hold:

$$U_{r,L,N-L}(L, t_0) \geq (L - 1)C_0 \quad (32)$$

Otherwise, the last finish time cannot reach t_0 , and T_L can be expressed as equation (24).

b) $L < L' + 1$

In this case, we have $T_L' < L't_a < t_0$. Fig. 5(c) illustrates the bandwidth allocation strategy. In order to make the first L peers finish download within t_0 , the following inequality must hold:

$$C_0 \leq \frac{1}{L} \left[C_0 + \left(1 - \frac{1}{L}\right) U_{r,N-L}(L, t_0) + U_{r,L}(L, t_0) \right] \quad (33)$$

Substituting equation (29) into inequality (33), we can deduce the following inequality:

$$C_0 \leq (1 - \varphi)F(L) \quad (34)$$

c) $L > L' + 1$

In this case we have $T_L' > (L' + 1)t_a > t_0$, therefore the discussions and equations to calculate T_L are the same as in case i) of a).

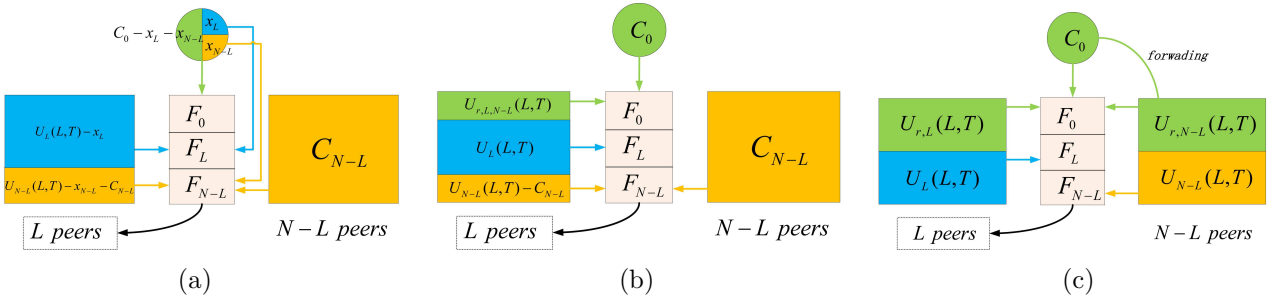


Fig. 5: Bandwidth allocation strategies (*differentiated service*). In (a) and (b), the last $N - L$ peers need not to forward F_0 from the source to the first L peers, while the opposite is true in (c).

Theorem 1: Suppose $L't_a < t_0 \leq (L' + 1)t_a$ ($L' \in [1, N - 1]$), if inequality (34) is true when $L = l + 1$, where $l + 1 \leq L'$, it is also true when $L \leq l$; if inequality (32) is true when $L = L' + 1$, inequality (34) is true when $L \leq L'$.

Proof: The proof of Theorem 1 is in the appendix.

Based on the above conclusions, we can generalize the *Multiplicity Theorem* [15] to the following φ -*Multiplicity Theorem*.

Theorem 2 (φ -*Multiplicity Theorem*): For a given P2P system, if the initial data distribution satisfies the *UCP condition* and the initial data ratio is φ , it is possible to let the first M peers finish their download in the bottleneck time t_0 . We define the largest integer M as the φ -*Multiplicity*, and it can be calculated as follows:

- If $t_0 < t_a$, no peer can finish its download in t_0 , then $M = 0$.
- If $t_0 = t_a$, there is one and only one peer can finish its download in t_0 , then $M = 1$.
- If $L't_a < t_0 \leq (L' + 1)t_a$ ($L' \in [1, N - 1]$), there are two cases: If inequality (32) is true when $L = L' + 1$, we have $M = L' + 1$; otherwise, M equals to the largest integer L no more than L' such that inequality (34) is true.

Fig. 6 illustrates the φ -Multiplicity Theorem in terms of the relation between t_0 and t_a (or the variation of φ).

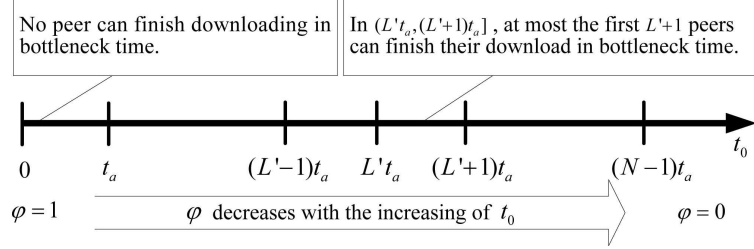


Fig. 6: Illustration of the φ -Multiplicity Theorem

Similar to section 2.3, we can calculate T_L as follows according to the φ -Multiplicity Theorem.

1) $L \leq M$

In this case, T_L must be constrained by the bottleneck time t_0 , i.e. $T_L = t_0$.

2) $L > M$

a) If $T_0 \leq Lt_a$, T_L is calculated by equation (24).

b) If $T_0 > Lt_a$, T_L is calculated by equation (31).

4 Numerical results

Suppose there are 18 peers with upload bandwidth 10, 10, 9, 9, 8, 8, 7, 7, 6, 6, 5, 5, 4, 4, 3, 3, 2, 2 (KB) respectively and one source peer distributing a file with the size of 1000KB. Fig. 7 gives the numerical results of both the original equal service and differentiated service models (i.e. $\varphi = 0$) and our extended models (i.e. $\varphi \in (0, 1]$).

We examine T_L under φ and L in these models. Fig. 7 shows that, the magnitude of φ -Multiplicity decreases with the increasing of φ . For a given φ , T_L remains the same when L is smaller or equal to the φ -Multiplicity, and increases rapidly with the increasing of L when L is larger than the φ -Multiplicity.

It is worthwhile to note that a smaller φ will seriously limit the utilization of peers' upload bandwidth, especially when L is small, hence will lead to a larger T_L . This is because when φ is small, it will take the source peer a long time to distribute its unique data F_0 to other peers. On the other hand, when L is large, all of peers' upload bandwidth will be better utilized, the increasing of φ will have slight influence on T_L .

5 Conclusions

In this paper, we modeled both the equal service process and the differentiated service process when peers' initial data distribution satisfies some special conditions (the *UCP condition*), and derived exact expression of the minimum last finish time. Both the *UCP condition* and the proposed models based on it have great theoretical and practical value.

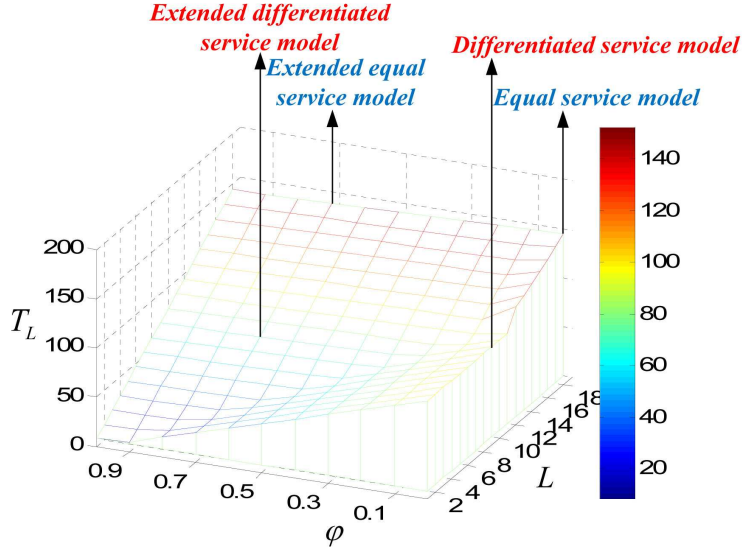


Fig. 7: *Equal service and differentiated service models.* ($\varphi \in [0, 1]$)

- The intrinsic relations among the initial data amount, the size of peer set and the minimum last finish time are revealed. Simply speaking, if the total data amount in all peers is small, the minimum last finish time is constrained by the source and cannot be further shortened by reducing the size of the first peer set. On the contrary, if the total data amount is large, the minimum last finish time can be much smaller than in the former case, and fewer peers can reach it.
- After the first peer set finishes its download, the data distribution of the second peer set still satisfies the *UCP condition*. Therefore, we can further use our proposed extended models to provide equal service or differentiated service to the second peer set. Furthermore, we can provide arbitrary degree of differentiated service to a certain number of peers. Suppose there are k sets of peers, then we can provide differentiated service from the first peer set to the k -th peer set in turn, the whole download process can be decomposed into $k - 1$ nested differentiated service processes and one equal service process, whichever process can be characterized by our extended differentiated service model or equal service model.

Due to there are no limits for peers' upload bandwidth and the size of each peer set, our proposed models have wide applicability and high extensibility for peer-to-peer file sharing.

Acknowledgement

This work is supported by the Next Generation Internet special of NDRC (Grant No. CNGI2008–112) and the National Key Technology R&D Program (Grant No. 2009BAH43B02).

Appendix

In this appendix we proof Theorem 1.

Let $U_{r,L,N-L}'(L, t_0)$ denote $(1 - \frac{1}{L})U_{r,N-L}(L, t_0) + U_{r,L}(L, t_0)$ in inequality (33). Define a function $ur(l)$ such that:

$$ur(l) = U_{r,l,N-l}'(l, t_0)/(l - 1) \quad (35)$$

By simple calculation we have $ur(l) = \frac{U}{l-1} - \frac{F_a}{t_0} - \frac{C_{N-l}}{l(l-1)}$, then:

$$ur(l+1) - ur(l) = \frac{C_{N-l} - U}{l(l-1)} - \frac{C_{N-(l+1)}}{l(l+1)} < 0 \quad (36)$$

If inequality (34) is true when $L = l + 1$, we have $ur(l+1) \geq C_0$, therefore $ur(l) > C_0$ holds, i.e. inequality (34) is true when $L = l$. Obviously, it is also true when $L < l$.

$U_{r,L,N-L}(L, t_0)$ can be rewritten as $U_{r,L,N-L}(L, t_0) = U - \frac{(L-1)F_a + F_{N-L}}{t_0}$ by definition. Since $t_0 \leq (L' + 1)t_a$, we have:

$$U_{r,L'+1,N-(L'+1)}(L' + 1, t_0) \leq U_{r,L'+1,N-(L'+1)}'(L' + 1, t_0) \quad (37)$$

Combining (35) - (37), we have $U_{r,L'+1,N-(L'+1)}(L' + 1, t_0)/L' < U_{r,L',N-L'}'(L', t_0)/(L' - 1)$. If inequality (32) is true when $L = L' + 1$, we have $U_{r,L'+1,N-(L'+1)}(L' + 1, t_0)/L' \geq C_0$. therefore $U_{r,L',N-L'}'(L', t_0)/(L' - 1) > C_0$ holds, i.e. inequality (34) is true when $L = L'$. Obviously, it is also true when $L < L'$.

References

- [1] B. Cohen, "Incentives build robustness in BitTorrent," in *Proc. of 1st Workshop on Economics of Peer-to-Peer Systems*, 2003.
- [2] W.C. Liao, F. Papadopoulos and K. Psounis, "Performance analysis of BitTorrent-like systems with heterogeneous users," *Performance Evaluation*, vol. 64, no. 9-12, pp. 876-891, October, 2007.
- [3] L. Mekouar, Y. Iraqi and R. Boutaba, "A contribution-based service differentiation scheme for peer-to-peer systems," *Peer-to-Peer Networking and Applications*, vol. 2, no. 2, pp. 146-163, June, 2009.
- [4] J.L. Chiang, Y.Y. Tseng and W.T. Chen, "Interest-Intended Piece Selection in BitTorrent-like peer-to-peer file sharing systems," *Journal of Parallel and Distributed Computing*, vol. 71, no. 6, pp. 879-888, June, 2011.
- [5] X.Y. Yang and G. de Veciana, "Performance of peer-to-peer networks: Service capacity and role of resource sharing policies," *Performance Evaluation*, vol. 63, no. 3, pp. 175-194, March, 2006.
- [6] X.Y. Yang and G. de Veciana, "Service capacity of peer to peer networks," in *Proc. of IEEE Infocom 2004*, pp. 2242-2252, March 7-11, 2004.
- [7] D.Y. Qiu and R. Srikant, "Modeling and performance analysis of BitTorrent-like peer-to-peer networks," *Computer Communication Review*, vol. 34, no. 4, pp. 367-377, October, 2004.
- [8] F. Clevenot-Perronnin, P. Nain and K.W. Ross, "Multiclass P2P networks: static resource allocation for service differentiation and bandwidth diversity," *Performance Evaluation*, vol. 62, no. 1-4, pp. 32-49, October, 2005.
- [9] E.W. Biersack, P. Rodriguez and P. Felber, "Performance analysis of peer-to-peer networks for file distribution," in *Proc. of 5th International Workshop on Quality of Future Internet Services*, pp. 1-10, September 29 - October 1, 2004.

- [10] F. Esposito, I. Matta, D. Bera and P. Michiardi, "On the impact of seed scheduling in peer-to-peer networks," *Computer Networks*, vol. 55, no. 15 pp. 3303-3317, October, 2011.
- [11] G. Wu and T.C. Chiueh, "How efficient is BitTorrent?" in *Proc. of 2006 SPIE Multimedia Computing and Networking Conf.*, pp. O710-O710, January 18-19, 2006.
- [12] J. Munding, R. Weber and G. Weiss, "Optimal scheduling of peer-to-peer file dissemination," *Journal of Scheduling*, vol. 11, no. 2, pp. 1094-6136, April, 2008.
- [13] J. Munding, R. Weber and G. Weiss, "Analysis of peer-to-peer file dissemination," *ACM SIGMETRICS Performance Evaluation Review*, vol. 34, no. 3, pp. 12-14, December, 2006.
- [14] R. Kumar and K.W. Ross, "Optimal peer-assisted file distribution: single and multi-class problems," in *Proc. of 1st IEEE Workshop on Hot Topics in Web Systems and Technologies*, pp. 20-30, November 13-15, 2006.
- [15] M. Mehyar, W. Gu, S.H. Low, M. Effros and T. Ho, "Optimal strategies for efficient Peer-to-Peer file sharing," in *Proc. of 32nd IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1337-1340, April 15-20, 2007.
- [16] B. Fan, J.C.S. Lui and D. Chiu, "The design trade-offs of BitTorrent-like file sharing protocols," *IEEE/ACM Transactions on Networking*, vol. 17, no. 2, pp. 365-376, April, 2009.
- [17] S.K.C. Jonathan, O.K.L. Victor and L. King-Shan, "Performance comparison of scheduling algorithms for peer-to-peer collaborative file distribution," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 1, pp. 146-154, January, 2007.
- [18] G.M. Ezovski, T. Ao and L.L.H. Andrew, "Minimizing average finish time in P2P networks," in *Proc. of IEEE Infocom 2009*, pp. 594-602, April 19-25, 2009.
- [19] C.H. Wen and R.C. Lin, "Towards minimizing average finish time of P2P file delivery under peer leaving," in *Proc. of 2010 International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, pp. 55-62, November 4-6, 2010.
- [20] Y. Liu, "Delay bounds of chunk-based peer-to-peer video streaming," *IEEE/ACM Transactions on Networking*, vol. 18, no. 4, pp. 1195-1206, August, 2010.
- [21] S. Sengupta, S. Liu, M. Chen, M. Chiang, J. Li and P.A. Chou, "Peer-to-Peer Streaming Capacity," *IEEE Transactions on Information Theory*, vol. 57, no. 8, pp. 5072 - 5087, August, 2011.