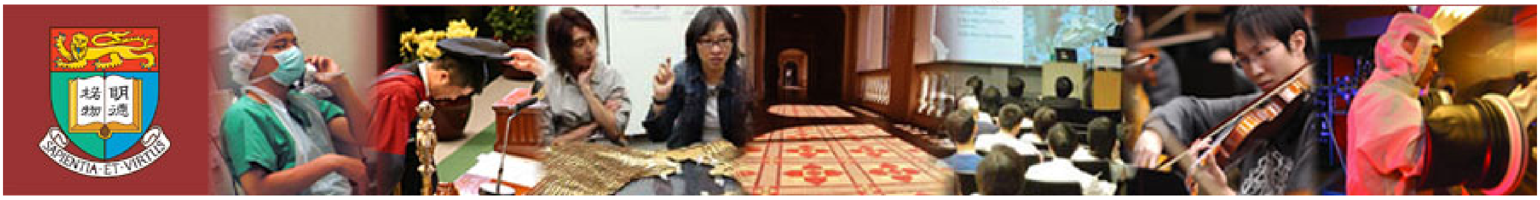

Novel Bandwidth Strategy for Wireless P2P File Sharing

Xiang Meng, Pui-Sze Tsang, King-Shan Lui
Department of Electrical and Electronic Engineering
The University of Hong Kong, Hong Kong
Email: {h0880103, tsangps, kslui}@eee.hku.hk

***Abstract*—With the rapid development of the mobile device technology and wireless network technology, the need of an efficient file sharing method on wireless network becomes more and more significant. Peer-to-Peer(P2P) file distribution, as a quite popular method being used now, is a promising choice. However, the limitation of bandwidth of wireless networks greatly restricts the performance of wireless P2P. In this paper, we propose a new idea of better utilizing the limited bandwidth to improve the file distribution performance. The criteria of an optimal splitting of the half-duplex bandwidth is deduced with mathematical analysis. To achieve a further improvement on the average distribution time, we also propose a grouping strategy which works with the bandwidth strategy. Simulation results show that our mechanism can efficiently reduce the file distribution time among wireless peers.**

I. Introduction

Ever since its appearance, Peer-to-Peer (P2P) has shown its superiorities on low resource requirement and high efficiency. It has gradually become the de facto way of massive file sharing. Nowadays, a large portion of the internet traffic is caused by P2P, and with the increasing attention on P2P, this portion will be continuously enlarged. On the other hand, given the rapid development of high speed wireless communication technologies, it is necessary to develop intelligent P2P protocols on wireless networks. Efficient file sharing over wireless P2P networks is one of the important issues. Unfortunately, existing P2P protocols may not be directly applied to wireless networks because the features of wireless links are different from those of a wired connection.

It is well-known that the available bandwidth on a wireless link is quite limited. Take the 3G network as an example, for commonly used WCDMA, it only supports 384Kbps in wide-area connection [1]. GPRS which is still popular in many developing countries only supports 171.2Kbps [2]. As a characteristic of wireless connection, surrounding environment can also reduce the actual bandwidth the users can get. Being a shared medium, nearby devices working in a similar frequency would further reduce the capacity available. These characteristics make bandwidth limitation becomes one of the major bottlenecks for P2P file sharing in wireless networks.

On the other hand, many wireless protocols are half-duplex[3] or time-division-duplex[4] in nature. In both duplex modes, transmission happens unidirectionally, with either upload or download but not both at the same time. Although the bandwidth allocated to upload and download can be adjusted in this case, it is not clear how to adjust the bandwidth to enhance the file sharing efficiency.

In this paper, we are interested in the scenario where each wireless peer is connected to a wireless access point (AP), and this AP allocates some fixed amount of bandwidth for the traffic, both upload and download, of this peer. Different APs can adopt different policies to allocate bandwidth and it is beyond the scope of this paper. Every connection between peers must go through the APs. The APs are connected through a backbone network which has abundant bandwidth. Therefore, the bandwidth bottleneck of a connection lies on the wireless link between the peer and its AP. We study how to efficiently share a file among these wireless users under the half-duplex capacity constraint.

The major performance metric we are using to measure the efficiency of a scheduling scheme is the time for all leechers to obtain the whole file, referred as the file distribution time in this paper. This metric reflects the overall performance. Another performance metric is the average distribution time which is used in [5]. It is the sum of finish times of all peers divided by the number of peers. This metric reflects the performance of individual peers to a certain extent. In this paper we seek answers to the following questions:

- What is the minimum time to distribute the file (minimum file distribution time) to all leechers?
- How to partition the capacity of each peer into upload and download portions to minimize the file distribution time?
- How to divide the peers into several groups to reduce the average distribution time?

The rest of the paper is organized as follows: Section II introduces some related work. Section III is a formal description of the problem. An optimal bandwidth partition is presented in section IV, while in section V, we introduce a grouping strategy which can work together with the optimal bandwidth partition. Before concluding in Section VII, supportive simulation result is represented in Section VI.

II. Related Work

The minimum file distribution time has been analyzed based on the fluid model by many researchers and there are several analytical models based on the *fluid mode*, in which every

978-1-61284-254-7/11/$26.00 ©2011 IEEE

bit in a file can be redistributed as soon as it is received. Mundinger et al. developed a closed-form expression for the minimum time required to distribute a file with every peer bears equal upload capacity and infinite download capacity[6]. The model is not a general one.

In [7], Kumar and Ross gave a general expression of the minimum file distribution time in a heterogeneous fluid model. However, they assumed that the upload and download capacities of each peer are independent and are known beforehand. In other words, they did not study how to allocate a bounded capacity into upload and download as in the wireless situation. To the best of our knowledge, there is no study on the minimum file distribution time in a wireless P2P file sharing system.

There are some protocols for P2P file sharing in wireless networks. [8] introduced a P2P file-sharing architecture for mobile networks which was based on the popular eDonkey protocol. They proposed additional caching entities and a crawler to make this P2P architecture able to reconcile the decentralized operation of P2P file sharing with interests of network operators. This work emphasized on making P2P protocols applicable in mobile wireless networks and had no discussion on the bandwidth arrangement. Authors in [9] studied a different problem. The authors proposed and compared several routing protocols for setting up end-to-end connections among peers in an ad hoc network for file sharing. The bandwidth partitioning issue of peers was not considered as well. The authors in [10] proposed a new protocol that enhances the fairness and energy efficiency, as well as provided an incentive mechanism. They focused on neighbor selection and the control of peers' status. Similar to other previous works, there was no discussion on bandwidth arrangement.

In summary, to the best of our knowledge, there is no work studying how to split the bounded capacity of a wireless link into upload and download portions so that the file distribution can finish as soon as possible. This motivates us to explore on this new direction of improvement.

Besides reducing the overall file distribution time, reducing the average distribution time is another popular direction for improving the P2P file sharing. We have developed some grouping algorithms [11][12] to reduce the average distribution time based on the theoretical studies developed by Kumar and Ross[7]. However, these protocols also assume that the upload and download capacities of a leecher are known. As far as we know, there is still no grouping scheme developed that considers also bandwidth arrangement.

III. Problem Description

A. Background

The file distribution time when the upload and download bandwidths are known has been studied by Kumar and Ross [7].

A leecher is a peer who wants to download a file. We denote the set of leechers as L and each leecher as l_i where $i = 1, ..., |L|$. The set of seeds S possesses the file and its total upload capacity is $u(S)$. All leechers have to download the whole file at the completion of the file distribution and do not possess any portion of the file in the beginning. The download capacity of l_i is $d(l_i)$ while the upload portion is $u(l_i)$. For convenience, we denote $u(L)$ as the sum of $u(l_i)$ of all leechers. The smallest $d(l_i)$ among all the leechers is d_{min}. The size of the file we want to distribute is F.

Based on a static fluid model which assumes that no peers join or leave during the distribution process, Kumar and Ross analyzed the minimum possible file distribution time that can be achieved. We refer this time as the minimum distribution time, and denote it as $T_{min}(S, L, F)$. It can be computed as follows:

$$T_{min}(S, L, F) = \max\{\frac{F}{d_{min}}, \frac{|L|F}{u(S)+u(L)}, \frac{F}{u(S)}\} \qquad (1)$$

In this expression, the first term $\frac{F}{d_{min}}$ is the time for the leecher with the minimum download bandwidth to download the whole file F; the second term $\frac{|L|F}{u(S)+u(L)}$ represents the time needed to distribute $|L|$ copies of F with all the upload bandwidth utilized for both seeds and leechers; the third term $\frac{F}{u(S)}$ shows the time required for seeds to provide one copy of F. Kumar and Ross also proposed an optimal scheduling algorithm, referred as the KR-algorithm in this paper, to realize this minimum distribution time which allows every peer finishes downloading the file at the minimum file distribution time.

Nevertheless, their work cannot be applied to wireless networks file sharing directly because the upload and download capacities of a wireless leecher are not given. In fact, the capacities can be adjusted and how to allocate the bandwidth would affect the distribution time. We aim at studying how to arrange the bandwidth to facilitate an efficient file sharing process.

B. Reducing the File Distribution Time

Based on the half-duplex communication assumption, we let the total capacity of upload and download of leecher l_i be c_i.

Given $u(S)$ and the set of leechers L, we want to determine the values of $d(l_i)$ and $u(l_i)$ of the leechers such that $d(l_i) + u(l_i) \leq c_i$ for every $l_i \in L$ while minimizing the distribution time. After dividing the capacity of each leecher into upload and download, we can apply Eq. (1) to compute the minimum distribution time of that particular partition scheme. Our goal is to identify the one that gives the smallest minimum distribution time among all partition schemes.

To understand how to divide the bandwidth optimally, we first study how the partition affects the value of each term in Eq. (1). The last term, $\frac{F}{u(S)}$, is independent of $u(l_i)$ and $d(l_i)$. Thus, how to arrange the leecher's available bandwidth makes no difference on this term. On the other hand, it is not likely that this term will be the minimum distribution time because the seed can contribute all its capacity to upload. Therefore, we focus our discussion on the situation where $\frac{F}{u(S)}$ is not the largest one among the three terms in Eq. (1). That is,

$$T_{min}(S, L, F) = \max\{\frac{F}{d_{min}}, \frac{|L|F}{u(S) + u(L)}\} \qquad (2)$$

To simplify the presentation, we write $T_{min}(S, L, F)$ as T_{min} in the rest of the paper. As F, $u(S)$ and $|L|$ are fixed for a specific network configuration, the problem becomes how to arrange d_{min} and $u(L)$ to achieve the minimum T_{min}.

C. Reducing the Average Distribution Time

While the KR-algorithm can guarantee all the peers finish the download at the theoretical minimum distribution time, it does not optimize the *average* download time. Let $T(l_i)$ be the time needed for leecher l_i to get the file. Under the KR-algorithm, $T(l_i) = T_{min}$ for every leecher $l_i \in L$. That is, the average download time, $\frac{\sum_{l_i \in L} T(l_i)}{|L|}$, is also T_{min}. When there is a leecher with very limited capacity, a fast leecher will suffer as it cannot finish early by utilizing all its capacity.

To remedy the situation, grouping strategy has been developed [11], [12]. Leechers are partitioned into several groups that each group follows the KR-algorithm. As slower leechers are isolated from faster ones, the faster leechers can finish earlier that their finish times are smaller than T_{min}, and thus reduce the average download time. Nevertheless, these grouping strategies cannot be applied in the wireless situation directly since they assume the download and upload capacities of leechers are given. In this paper, we study how to partition the wireless leechers into different groups to reduce the average download time.

IV. Optimal Bandwidth Partition

In this section, we present how to divide c_i into $u(l_i)$ and $d(l_i)$ such that T_{min} is minimized. We also develop a formula for computing the minimum time to distribute a file among the leechers in L.

To minimize T_{min} of a certain network, we should make both d_{min} and $\frac{u(S)+u(L)}{|L|}$ as large as possible. Without loss the generality, we increase one of them first, say $\frac{u(S)+u(L)}{|L|}$. In this term, $u(S)$ and $|L|$ are fixed for a given network and we can only vary $u(L)$. For $u(L)$, we have a constraint that $u(L) \leq \sum_{l_i \in L}(c_i - d_{min})$. It means we can increase $u(L)$ in this range without affecting the value of d_{min}. In another words, once $u(L) = \sum_{l_i \in L}(c_i - d_{min})$, if we further enlarge $u(L)$, we will definitely cause a reduction of d_{min}. The goal is to make both terms as large as possible, and thus we should equating $\frac{F}{d_{min}}$ and $\frac{|L|F}{u(S)+u(L)}$, i.e. we need to satisfy Eq. (3) in case where $u(L) = \sum_{l_i \in L}(c_i - d_{min})$.

$$d_{min} = \frac{u(S) + u(L)}{|L|} \qquad (3)$$

$u(L) = \sum_{l_i \in L}(c_i - d_{min})$ implies that we should set all the leechers' download bandwidth to the lower boundary d_{min}. By substituting $u(L)$ in Eq. (3) with $\sum_{l_i \in L}(c_i - d_{min})$, we can get d_{min} which is:

$$d_{min} = \frac{u(S) + \sum_{l_i \in L} c_i}{2|L|} \qquad (4)$$

So the optimal configuration for download capacity is to set all $d(l_i)$ to be d_{min} which is decided by Eq. (4). With the previous configuration, we can get the optimal value for T_{min}, which is:

$$T_{min} = \frac{2F|L|}{u(S) + \sum_{l_i \in L} c_i} \qquad (5)$$

Nevertheless, Eq. (4) does not consider the constraint imposed by the capacities (c_i) of the peers. Let C_{min} be $min\{c_i | l_i \in L\}$. Since $d_{min} \leq C_{min}$, by limiting Eq. (4) to be less than C_{min}, we can get the requirement for Eq. (4) to be applicable, which is:

$$u(S) \leq \sum_{l_i \in L}(2C_{min} - c_i) \qquad (6)$$

When $u(S) > \sum_{l_i \in L}(2C_{min} - c_i)$, Eq. (2) shows that among d_{min} and $\frac{u(S)+u(L)}{|L|}$, the smaller one determines the final result. Thus our job is to find the smaller term and make it as large as possible.

By setting $d(l_i)$ to be d_{min} and given $d_{min} \leq C_{min}$, we have:

$$\begin{aligned} u(L) &= \sum_{l_i \in L}(c_i - d_{min}) \\ &\geq \sum_{l_i \in L} c_i - |L|C_{min} \end{aligned}$$

On the other hand,

$$\begin{aligned} u(S) &> \sum_{l_i \in L}(2C_{min} - c_i) \\ &= 2|L|C_{min} - \sum_{l_i \in L} c_i \end{aligned}$$

It implies

$$\begin{aligned} \frac{u(S)+u(L)}{|L|} &> \frac{\sum_{l_i \in L} c_i - |L|C_{min} + 2|L|C_{min} - \sum_{l_i \in L} c_i}{|L|} \\ &= C_{min} \\ &\geq d_{min} \end{aligned}$$

Therefore, $T_{min} = \frac{F}{d_{min}}$. To minimize T_{min}, we should set d_{min} to be C_{min}. Then,

$$T_{min} = \frac{F}{C_{min}} \qquad (7)$$

In summary, when the KR algorithm is used for scheduling the distribution, we should set $d(l_i)$ to be d_{min} for every leecher l_i where d_{min} can be found by (4) when (6) holds; otherwise, d_{min} should be C_{min}.

V. Group Strategy

In the previous section, we discussed how we can achieve the best T_{min} by adjusting the bandwidth. However, from the perspective of a single peer, it is not the best since everyone ends at the same time no matter how much bandwidth resource it has. Thus we introduce a grouping strategy, which does not prolong the overall distribution process but allows faster peers to finish the download earlier.

We want to divide the leechers into $N > 1$ groups and break the seed into N portions, one for each leecher group. We label the leecher groups as L_i and the seed portions as S_i, $1 \leq i \leq N$. Group G_i consists of L_i and S_i. Every leecher belongs to one and only one group. That is, a valid grouping should satisfy the following property:

- $L_i \cap L_j = \emptyset$ when $i \neq j$
- $\cup_{1 \leq i \leq N} L_i = L$
- $\sum_{1 \leq i \leq N} u(S_i) = u(S)$

We do not want to prolong the whole distribution process while reducing the average distribution time. As the KR-algorithm is used for scheduling the distribution, the download time of all the leechers in the same group would be the same. We denote the download time of group G_i as $T(G_i)$. Note that $T(G_i) = T_{min}(S_i, L_i, F)$ in Eq. (1). The average download time T_{avg} among the leechers becomes $\frac{\sum_{1 \leq i \leq N} |L_i| \times T(G_i)}{|L|}$. We aim at developing a grouping scheme to identify G_i such that $T_{avg} < T_{min}$ and $T(G_i) \leq T_{min}$ for all $1 \leq i \leq N$.

A. When is Grouping Not Beneficial

When $u(S) \leq \sum_{l_i \in L}(2C_{min} - c_i)$, we can't conduct a beneficial grouping. To reduce T_{avg}, at least one group must have a smaller file distribution time than the overall time T_{min}. In the following, we will show that it is not possible to be done without prolonging the overall file distribution time T_{min} under $u(S) \leq \sum_{l_i \in L}(2C_{min} - c_i)$.

Firstly, we assume the network is partitioned into two groups, G_1 and G_2. Since G_1 and G_2 are partitioned from the original network, we must have

$$\frac{u(S_1) + \sum_{l_i \in L_1} c_i + u(S_2) + \sum_{l_i \in L_2} c_i}{|L_1| + |L_2|} = \frac{u(S) + \sum_{l_i \in L} c_i}{|L|} \tag{8}$$

Assume we make $T(G_1) < T_{min}$. For the following two situations, one and only one must be true for G_1

$$\frac{u(S_1) + \sum_{l_i \in L_1} c_i}{|L_1|} > \frac{u(S) + \sum_{l_i \in L} c_i}{|L|} \tag{9}$$

$$\frac{u(S_1) + \sum_{l_i \in L_1} c_i}{|L_1|} \leq \frac{u(S) + \sum_{l_i \in L} c_i}{|L|} \tag{10}$$

We now argue that (10) can't be held when $T_{min}(G_1) < T_{min}$. When $u(S) \leq \sum_{l_i \in L}(2C_{min} - c_i)$, $\frac{u(S)+\sum_{l_i \in L} c_i}{|L|} \leq C_{min}$. Since C_{min} is the smallest capacity of the leechers in the whole network, the smallest capacity of the leechers in L_1, denoted as $C_{min}(G_1)$, must be at least C_{min}. From (10),

$$\begin{aligned} \frac{u(S_1) + \sum_{l_i \in L_1} c_i}{2|L_1|} &\leq \frac{u(S) + \sum_{l_i \in L} c_i}{2|L|} \\ &\leq C_{min} \\ &\leq C_{min}(G_1) \end{aligned}$$

Rearrange the above expression, we can get

$$u(S_1) \leq \sum_{l_i \in L}(2C_{min}(G_1) - c_i)$$

Based on the discussion in Section IV, when (6) holds, $T(G_1)$ can be found by (5). Then, we have $T(G_1) > T_{min}$, which contradicts with our earlier assumption that $T(G_1) < T_{min}$. We can conclude that when $u(S) \leq \sum_{l_i \in L}(2C_{min} - c_i)$ and $T(G_1) < T_{min}$, (9) must hold.

When (9) holds, by (8), we know

$$\begin{aligned} \frac{u(S_2) + \sum_{l_i \in L_2} c_i}{|L_2|} &< \frac{u(S) + \sum_{l_i \in L} c_i}{|L|} \\ &< \frac{u(S_1) + \sum_{l_i \in L_1} c_i}{|L_1|} \end{aligned}$$

This makes $T(G_2) > T_{min}$ and violates the requirement that $T(G_i)$ must be at most T_{min}.

Similar argument can be applied when $N > 2$. Therefore, we can conclude that it is not beneficial to perform grouping when $u(S) \leq \sum_{l_i \in L}(2C_{min} - c_i)$.

B. How To Partition Groups

The purpose of grouping is to make T_{avg} smaller. With a fixed total number of leechers, the only way to achieve it is to reduce the sum of the download time of all the leechers. As in the KR-algorithm, every leecher ends at the same time, the leechers who give a bad performance will slow down the whole group. If we can isolate these leechers, the rest of the leechers will most likely perform better.

Given a group of peers, among all the leechers, the leecher with the smallest capacity is most likely to slow down the download process. According to the previous discussion, we isolate this leecher by putting it into one group, the other leechers into another. We can continue to apply the method to further isolate slower leechers from faster ones, and the faster leechers can finish the file distribution earlier.

The optimal bandwidth strategy works on the assumption that among three terms in (1), $\frac{F}{u(S)}$ is not the largest term. To be compatible with the optimal bandwidth strategy, our grouping strategy also adopts the arrangement that for every single group, the total upload capacity from seeds is no smaller than the minimum total capacity of leecher, i.e. $u(S_i) \geq C_{min}(G_i)$ for $1 \leq i \leq N$. By doing so, we secure that $\frac{F}{u(S)}$ is always smaller or equal to $\frac{F}{d_{min}}$.

We now describe the details of our protocol as follows:

- Check whether $u(S) > \sum_{l_i \in L}(2C_{min} - c_i)$. If not, no further grouping is needed;
- Arrange the leechers according to non-decreasing order of their capacities. That is, $c_i \geq c_j$ if $i > j$.
- Assign $L_1 = \{l_1\}$ and $u(S_1) = c_1$. That is, we isolate l_1 from the others and assign enough seed capacity to its group. We apply the same isolation to other leechers until the seed capacity is not enough to make the last group satisfy $u(S_i) \geq C_{min}(G_i)$. Formally,

1) Find N where $\sum_{1\leq i\leq(N+1)} c_i > u(S)$ but $\sum_{1\leq i\leq N} c_i \leq u(S)$.
2) For $1 \leq i \leq N-1$, assign $L_i = \{l_i\}$ and $u(S_i) = c_i$, while $L_N = \{l_N, ..., l_{|L|}\}$ and $u(S_N) = u(S) - \sum_{1\leq i\leq(N-1)} c_i$

- If $T(G_N)$ is smaller than $T(G_{N-1})$, the grouping process stops. Otherwise, we merge G_{N-1} and G_N into a new group, and the number of groups becomes $N-1$;
- Repeat the previous step until the grouping is finished.

Our grouping will not prolong the overall file distribution time, as well as satisfies the two arrangements, which are isolating the slower leechers and making $u(S_i) \geq C_{min}(G_i)$ for all groups as we stated at the beginning of this section.

After grouping is done, we can utilize (11) to calculate the time for G_i when $1 \leq i \leq N-1$.

$$T(G_i) = \frac{F}{c_i} \tag{11}$$

For the last group G_N, we can directly apply (5) to calculate $T(G_N)$ when G_N satisfies (6), or apply (7) when (6) doesn't hold. Under our grouping scheme, $u(S_N)$ is assigned after ensuring enough $u(S_i)$ for $1 \leq i \leq (N-1)$, thus in most cases, $u(S_N)$ is usually not sufficient to exceed the boundary of (6). As a result, (5) is most likely to be taken when calculate $T(G_N)$.

We now present an example to illustrate our grouping scheme. Table I gives the capacities of leechers in a network with $u(S) = 1000kbps$. The file size is set to be $F = 100Mb$.

TABLE I
TOTAL CAPACITIES OF THE LEECHERS

leechers	l_1	l_2	l_3	l_4	l_5
$c_i/kbps$	50	70	100	120	150
leechers	l_6	l_7	l_8	l_9	l_{10}
$c_i/kbps$	200	250	300	400	500

First, we should proceed to perform the grouping because $u(S) > \sum_{l_i\in L}(2C_{min} - c_i)$. The initial grouping becomes:

$$\begin{array}{r|l|l}
G_1: & l_1 = 50 & u(S_1) = 50 \\
G_2: & l_2 = 70 & u(S_2) = 70 \\
G_3: & l_3 = 100 & u(S_3) = 100 \\
G_4: & l_4 = 120 & u(S_4) = 120 \\
G_5: & l_5 = 150 & u(S_5) = 150 \\
G_6: & l_6 = 200 & u(S_6) = 200 \\
G_7: & l_7 = 250, l_8 = 300, l_9 = 400, l_{10} = 500 & u(S_7) = 310
\end{array}$$

Then applying the optimal bandwidth strategy to G_7, we can get $T(G_7) = 100k/220 = 454.55$. This time is smaller than that of G_6, which is $T(G_6) = 100k/200 = 500$, so the process ends. The overall file distribution time for this network is $T_{min} = T(G_1) = 100k/50 = 2000$, which is the same with the minimum file distribution time before applying grouping. The T_{avg} is improved to

$$\begin{aligned} T_{avg} &= [100k/50 + 100k/70 + 100k/100 + 100k/120 \\ &\quad +100k/150 + 100k/200 + (100k/220) \times 4]/10 \\ &= 824.68, \end{aligned}$$

which is less than half of the average time without grouping.

VI. Simulation result

In this section, we present the simulation result to illustrate the improvement of our optimal bandwidth strategy and grouping strategy.

For a comprehensive consideration, the simulation is done with different combinations of seeds and leechers. The number of leechers is picked from $[50, 60, 70, 80, 90, 100, 110]$, with the capacities of peers selected within $[171.2Kbps, 2Mbps]$. The lower boundary of this range is the data transmission rate for GPRS[2], and the upper boundary is the data transmission rate of WCDMA[1]. For the capacity of seed, we selected it from $k \times [171.2Kbps, 2Mbps]$, where k can be $[5, 10, 15, 20, 25]$. As k increases, the capacity of seed also increases. For each combination, we generate 30 sets of configurations. We follow a random process when selecting the capacities of peers and seed in every data set, and every point in the figures is the average of the 30 configurations under the same network setting.

First, we present the improvement of the optimal bandwidth strategy. We compare our optimal bandwidth strategy with a random bandwidth strategy. Under the random scheme, the upload capacity of l_i is selected randomly from 10% to 90% of c_i, while the remaining capacity is all used for download. We present the improvement by showing the ratio $r = \frac{\text{distribution time of the random scheme}}{\text{distribution time of our optimal scheme}}$. The larger the ratio, the better our scheme is.

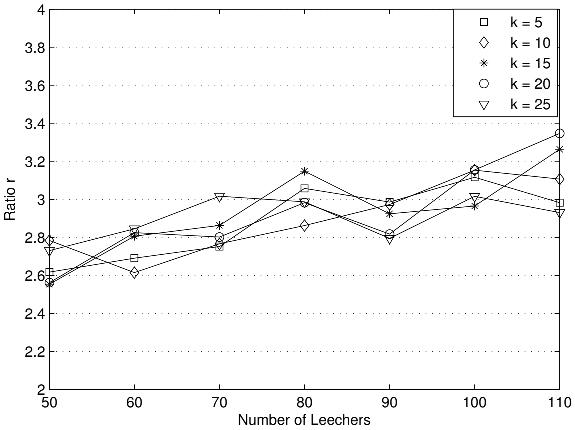

Fig. 1. Comparison of optimal and random bandwidth strategy

Fig.(1) shows the simulation result of the comparison. From the figure, we can see that nearly all the ratios are larger than 2.6. The optimal bandwidth strategy does improve the file distribution time greatly.

Then we present the improvement on T_{avg} after applying our grouping strategy. We define another ration $p = \frac{\text{average distribution time without grouping}}{\text{average distribution time with grouping}}$. Under the KR-algorithm, every leecher in a group finishes at the same time, so the average distribution time without grouping is actually equals to T_{min} of that data set.

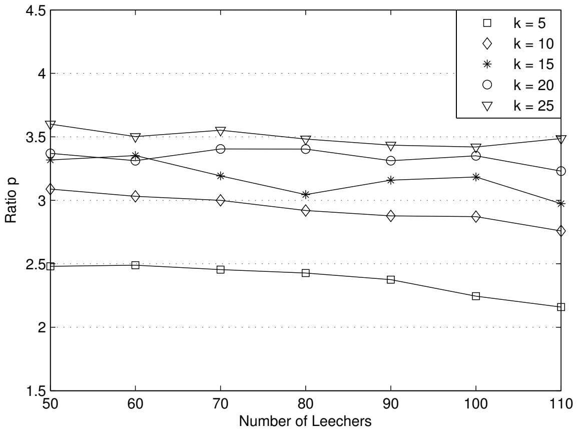

Fig. 2. Improvement on the average

Fig.(2) shows the simulation result of the improvement of the grouping strategy. As we can see, all the ratios are larger than 2, which means by applying our grouping strategy, we can reduce the T_{avg} at least by half.

We can also observe an obvious ascending corresponding relationship between the ratio and k, as well as a slightly decrease of the ratio with the increase of the number of leechers. With more seed capacity, we can have more groups. Following our grouping strategy, the more groups there are, the better the final result will be since more slower leechers are isolated. The observation conforms the analysis. For the slightly decrement following the increase of the leechers, it can be explained by analyzing the last group. Following our grouping strategy, $T(G_N)$ is most likely decided by (5), when the number of leechers in this group increases, $\frac{u(S_N)+\sum_{N\leq i\leq |L|} c_i}{2|L_N|}$ will most likely be decreased. So $T(G_N)$ is likely to increase. Nevertheless, as only one group experiences this phenomenon, the trend is not significant.

VII. Conclusion

In this paper, we illustrated and proved the idea that by arranging the bandwidth properly, we could achieve significant improvement for wireless P2P networks. We gave a complete analysis on optimizing the bandwidth configuration of P2P networks based on Kumar and Ross's work and derived a set of criteria accordingly. For a further consideration of the average download time of the network, we gave a grouping strategy which was compatible with the optimal bandwidth strategy. At the end of the paper, we conducted a series of simulations to show the superiorities of our work.

References

[1] Stefan Parkvall, Erik Dahlman, Pal Frenger, Per Beming, and Magnus Persson, "The high speed packet data evolution of wcdma," in *Proc. of the 12th IEEE Personal, Indoor and Mobile Radio Communications*, 2001.

[2] J Gozalvez and J Dunlop, "High-speed simulation of the gprs link layer," in *Proc. of the 11th IEEE Personal, Indoor and Mobile Radio Communications*, 2000.

[3] Pejman Roshan and Jonathan Leary, *802.11 Wireless LAN fundamentals*, Cisco Press, first edition, 2005.

[4] R. Esmailzadeh, M. Nakagawa, and E. A. Sourour, "Time-division duplex cdma communications," *IEEE Personal Communications*, vol. 4, pp. 51–56, 1997.

[5] G. M. Ezovski, Ao Tang, and Lachlan L.H. Andrew, "Minimizing average finish time in p2p networks," in *Proc. of the INFOCOM 2009*, pp. 594–602.

[6] G. Mundinger, R. Webb, and G. Weiss, "Analysis of peer-to-peer file dissemination," in *Proc. of the 8th Workshop on MAthematical performance Modeling and Analysis*, 2006.

[7] R. Kumar and K. Ross, "Peer assisted file distribution: The minimum distribution time," in *Proc. of the 1st IEEE Workshop on Hot Topics in Web Systems and Technologies*, 2006.

[8] Jens O. Oberender, Frank-Uwe Andersen, Hermann de Meer, Ivan Dedinski, Tobias Hoßfeld, Cornelia Kappler, Andreas Mäder, and Kurt Tutschku, "Enabling mobile peer-to-peer networking," in *Proc. of the 1st International Workshop of the EURO-NGI Network of Excellence*, 2004.

[9] G. Ding and B. Bhargava, "Peer-to-peer file-sharing over mobile ad hoc networks," in *Proc. of the 2nd IEEE Pervasive Computing and Communications Workshops*, 2004.

[10] A.K.-H. Leung and Y.-K. Kwok, "On localized application-driven topology control for energy efficient wireless peer-to-peer file sharing," in *IEEE Transactions on Mobile Computing*, 2008, pp. 66–80.

[11] Lingjun Ma, Pui-Sze Tsang, and King-Shan Lui, "Improving file distribution performance by grouping in peer-to-peer networks," in *IEEE Transactions on Network and Service Management*, September 2009, vol. 6, Issue 3, pp. 149–162.

[12] Pui-Sze Tsang, Xiang Meng, and King-Shan Lui, "A novel grouping strategy for reducing average distribution time in p2p file sharing," in *IEEE International Conference on Communications*, 2010.