

Maintaining Comprehensive Resource Availability in P2P Networks ^{*}

Bin Xiao¹, Jiannong Cao¹, and Edwin H.-M. Sha²

¹ Department of Computing
Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong
{csbxiao, csjcao}@comp.polyu.edu.hk

² Department of Computer Science
University of Texas at Dallas
Richardson, Texas 75083, USA
edsha@utdallas.edu

Abstract. In this paper, the resource availability issue has been addressed with respect to the intermittent connectivity and dynamic presence of peers in a P2P system. We aim to maintain more comprehensive resource available. The contribution of this paper is two fold. First, we analyze the probability of a peer to become a seed consisting of all content. Second, a simple, distributed comprehensive algorithm for resource management is proposed. In order to increase the availability of resources in a P2P network, the peer as well as the fragment of shared content are deliberately allocated during the content sharing process.

1 Introduction

P2P (peer-to-peer) systems have drawn much attention in recent years. The infrastructure of a P2P network makes a peer to act as both a server and a client. A peer (e.g., a computer in a network) can build a direct connection with another one without the involvement of a server. Such a network architecture avoids the drawbacks of conventional client-server model, such as the computation bottleneck on servers, and thus greatly improved the efficiency of system computing performance [6]. The P2P computing encompasses applications in a P2P network that can be categorized as distributed computing, content sharing and collaboration [2, 7]. The popularity of content sharing (or file sharing) systems (e.g., Gnutella [8], Napster and Freenet [3]) in turn encourages the continuous research for advanced technologies in P2P networks.

Sharing content [5] in the P2P context addresses the techniques and technologies for effective ways of discovering the existence and locations of content, and furthermore getting it transferred. The shared content can be files or a storage. For the more popular file sharing applied in Internet, the content can be

^{*} This work is partially supported by HK Polyu ICRG A-PF86 and COMP 4-Z077, and also by TI University Program, NSF ETA 0103709, Texas ARP 009741-0028-2001 and NSF CCR-0309461

one or several files. During a file sharing session for a task, a peer holding part of content would receive data from and send data to others in a P2P network. However, a peer containing complete content only takes part in uploading data to others, which is referred to a *seed* in the paper. The success for a peer in a file sharing session is identified by the file totally incorporated in its local storage. The shortage of a small piece of content can make its previous work become useless since the integrity of the file is not reached. Much work has been done as how to efficiently discover the required resources (e.g., particular files) [1, 9, 11]. However, further investigation on the content sharing reveals the equal importance of complete resource available in P2P networks.

Upon receiving various requests in a P2P network, a peer faces the challenge as how to assign its resources. For instance, which peer would get service and which fragment of the shared content would be delivered first. An effective resource assignment method can yield more seeds within a constant time for a P2P infrastructure. Large number of peers with the total content lead to the resource availability increased, and much more fault tolerance when some seed peers depart. Furthermore, the distribution of generated seeds can be more balancing inside the whole system. The network traffic in terms of content transferring is evenly distributed since the data can be delivered from many places. Thus, the resource availability issue for content sharing is addressed in this paper in order to maintain more comprehensive resources in a P2P network.

The contribution of this paper is two fold. First, we analyze the probability of a peer to become a seed consisting of all content. The probability is formalized by the duration time of a peer in a P2P network modeled by uniform, exponential and normal random variables respectively. Second, a simple, distributed comprehensive algorithm for resource management is proposed. This algorithm can be deployed in each peer only with local information known inside a P2P system. In order to achieve more available resources, the peer as well as the pieces of content are deliberately allocated during the content sharing process in the described algorithm. We believe that the presented method can outperform randomized resource assignment techniques in most cases, such as longer session time, large number of peers that successfully contain all content.

In the next section, an example will be given to illustrate the request-based content sharing in a P2P system. The state of a peer is denoted by a 6-tuple. In Section 3, after the presented model for content request, the probability for a peer to become a seed is analyzed. The duration time of a peer is formulated by continuous uniform, exponential and normal random variables respectively. The comprehensive resource management algorithm is described in detail in Section 4. Concluding remarks are given in Section 5.

2 Request-based Content Sharing

A P2P network can be represented by an undirected graph $G = (V, E)$ where each peer is denoted by a node $v \in V$. An edge $(u, v) \in E$ represents a bidirectional communication link between peer u and v . Let the size of a shared

content be f_s . In this paper, we concern the sharing of a file unit F among a given P2P system in a session. For each node $v \in V$, its bandwidth resource for data transfer is limited. A 6-tuple $(B_{uu}, B_{ul}, B_{du}, B_{dl}, f_l, t_p)$ is associated to denote its current state, where B_{uu} describes the bandwidth used for node v to upload data in F while B_{ul} represents the left bandwidth, B_{du} shows the occupied bandwidth to download F while B_{dl} means the left bandwidth for content downloading, f_l is the size of the left content, and t_p provides the time passed for peer v connected in the system in a session of content sharing of F . The size of f_l is calculated by the number of missed blocks of file F , which is denoted by the percentage of f_s . It is possible that node v is downloading file F from other peers while uploading pieces of F concurrently in a distributed P2P system. For a node v with $f_l = 0$, which means peer v contains the whole file F , v is defined as a *seed* in the system. The seed node only sends data in a particular file F to non-seed nodes. For those nodes that are with $f_l \neq 0$, they are attempting to derive data fragment to minimize f_l and as a source to provide data to others.

A P2P system is shown in Figure 1 for the sharing of a file F among 5 peers. The network topology represents the request connections in a file sharing session. Along with each engaged node, a table provides its current state. The state information is free to others whenever a connection is built. For node A , its $f_l = 0$ reveals that A can be treated as a seed. Nodes B, C and D are in the data transmission process and part of the content is attained. Node E is at the beginning of file downloading since f_l is the same as f_s . Suppose that the uploading bandwidth of A is free of occupation ($B_{uu} = 0$). Upon the request from node B and C of sharing of file F , node A can clearly view their tables as in Figure 1. Thus, node A needs to make a decision as how to assign its limited bandwidth for package delivery to those 2 peers (B and C). Moreover, peers, such as B, C and D , face the same problem to handle requests from others.

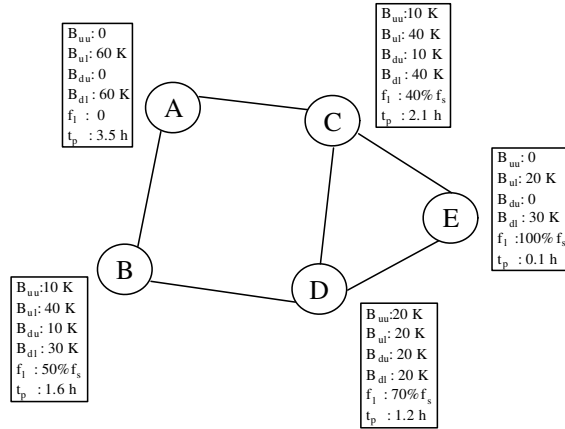


Fig. 1. The state of a file sharing in a P2P system.

3 Probability Analysis

3.1 A Model for Content Request

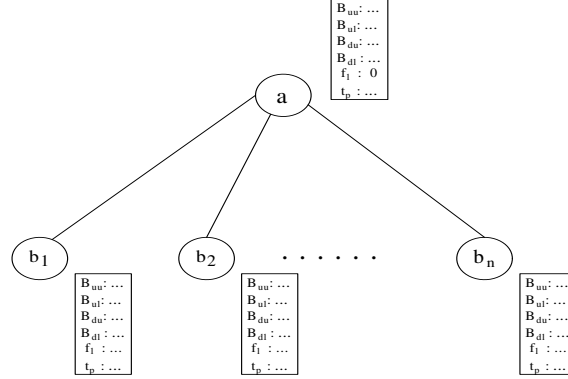


Fig. 2. n peers send their requests to peer a .

In Figure 2, a graph depicts the case as n peers (from b_1 to b_n) send their requests as content sharing to peer a in a P2P distributed system. Peer a is assumed to be a seed that consists of the entire content. Although a P2P setting is loosely organized and fully decentralized, a participant can be aware of the configurations of those peers connecting to it. Hence, graph in Figure 2 to peer a is transparent after message exchange.

More seeds available in a P2P system, it is more fault tolerance. Given that peer b_i is supported by peer a , let B_i ($i = 1, 2, \dots, n$) be the event that peer b_i becomes a seed. The value of B_i is either 1 or 0 where 1 represents the defined event to be true. Let t_i be the time interval from its request to peer a to the time that B_i is true. Thus, we have

$$t_i = \frac{f_{l_i}}{\min(B_{ul_a}, B_{dl_i}) + B_{du_i}}$$

where the subscript i denotes the parameter of peer b_i while a for peer a in this paper. The content obtained by peer b_i can either from other peers, which is represented as through a channel by B_{du_i} bandwidth (assumed to be consistently existing), or from the seed a . The second path with the source node a has the maximum bandwidth supported by $\min(B_{ul_a}, B_{dl_i})$ for packet delivery. f_{l_i} means the content contained by a but not in peer b_i (even if peer a is not a seed, the above equation is valid following the same definition of f_{l_i}). Let T_a, T_{b_i} be the event that peer a, b_i will last t_i time longer respectively. Then

$$B_i = T_a \cdot T_{b_i} \tag{1}$$

In a P2P distributed system, peer a and peer b_i can be assumed to be two independent hosts. It follows that T_a and T_{b_i} are two independent events. From Equation 1, the probability of event B_i to be true can be depicted as

$$P(B_i) = P(T_a) \cdot P(T_{b_i}) \quad (2)$$

Let X_a be the random variable that denotes the time for peer a online. Because peer a has already been in the session for a t_{p_a} time period, the probability of event T_a to be true is under the conditional probability case. Hence, we have:

$$P(T_a) = \frac{P\{X_a > t_{p_a} + t_i\}}{P\{X_a > t_{p_a}\}} \quad (3)$$

Let X_{b_i} be the random variable that denotes the time for peer b_i online. From Equation 2 and 3, we can rewrite $P(B_i)$ as

$$P(B_i) = \frac{P\{X_a > t_{p_a} + t_i\}}{P\{X_a > t_{p_a}\}} \cdot \frac{P\{X_{b_i} > t_{p_{b_i}} + t_i\}}{P\{X_{b_i} > t_{p_{b_i}}\}} \quad (4)$$

3.2 Uniform Random Variable

Although the topology of a P2P network is constantly changed, after an initial setting up process for content sharing of a particular file unit, the network can be treated as in a steady state. In other words, the number of peers arrived equalizes those departed and the network keeps in a balanced situation. Let X be a uniform random variable to denote the connecting time for a peer on the interval (α, β) . Its probability density function is given by

$$f(x) = \begin{cases} \frac{1}{\beta-\alpha}, & \text{if } \alpha < x < \beta \\ 0, & \text{otherwise} \end{cases}$$

For a given $t \in (\alpha, \beta)$, $P\{X > t\} = 1 - P\{X \leq t\} = 1 - \frac{t-\alpha}{\beta-\alpha} = \frac{\beta-t}{\beta-\alpha}$. Thus

$$P(B_i) = \frac{\beta - t_{p_a} - t_i}{\beta - t_{p_a}} \cdot \frac{\beta - t_{p_{b_i}} - t_i}{\beta - t_{p_{b_i}}} \quad (5)$$

3.3 Exponential Random Variable

The duration time for a peer stays connecting to a network can be deployed by the exponential random variable with parameter λ . It is said the behavior of a P2P system can be better modeled by a stochastic and memoryless setting [10]. The arrival and departure of peers comply with a Poisson process. Each peer is independently and exponentially distributed online for content sharing. The probability density function for an exponential random variable X with parameter λ is given by:

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases}$$

The memoryless property of the exponential random variable makes that the probability of a peer surviving for another t time longer does not involve any previous performance. Thus

$$P(B_i) = P\{X_a > t_i\} \cdot P\{X_{b_i} > t_i\} = e^{-\lambda t_i} \cdot e^{-\lambda t_i} = e^{-2\lambda t_i} \quad (6)$$

3.4 Normal Random Variable

Suppose that the connection time for a peer in a P2P system approximately satisfies the normal distribution with parameters μ and σ^2 . Let X be the modified normal random variable to represent the duration time of a peer and the probability density function can be represented as following

$$f(x) = \begin{cases} \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2} & x \geq 0 \\ 0 & x < 0 \end{cases}$$

Thus, we have

$$\begin{aligned} P(X > t) &= 1 - P(X \leq t) = 1 - \frac{1}{\sqrt{2\pi}\sigma} \int_0^t e^{-(x-\mu)^2/2\sigma^2} dx \\ &= 1 - \frac{1}{\sqrt{2\pi}\sigma} \int_0^\mu e^{-(x-\mu)^2/2\sigma^2} dx - \frac{1}{\sqrt{2\pi}\sigma} \int_\mu^t e^{-(x-\mu)^2/2\sigma^2} dx \\ &= 1 - \frac{1}{\sqrt{2\pi}\sigma} \int_0^\mu e^{-y^2/2\sigma^2} dy - \frac{1}{\sqrt{2\pi}\sigma} \int_0^{t-\mu} e^{-y^2/2\sigma^2} dy \quad (\text{let } y = x - \mu) \\ &= 1 - \frac{1}{\sqrt{\pi}} \int_0^{\mu/\sqrt{2}\sigma} e^{-t^2} dt - \frac{1}{\sqrt{\pi}} \int_0^{(t-\mu)/\sqrt{2}\sigma} e^{-t^2} dt \quad (\text{let } t = \frac{y}{\sqrt{2}\sigma}) \\ &= 1 - \frac{\text{erf}(\mu/\sqrt{2}\sigma)}{2} - \frac{\text{erf}((t-\mu)/\sqrt{2}\sigma)}{2} \end{aligned}$$

where $\text{erf}(z)$ is an ‘‘error function’’ from the normal distribution, which is defined as $\text{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt$ [4]. Hence, from Equation 4 it follows

$$P(B_i) = \frac{1 - \frac{\text{erf}(\mu/\sqrt{2}\sigma)}{2} - \frac{\text{erf}((t_{p_a} + t_i - \mu)/\sqrt{2}\sigma)}{2}}{1 - \frac{\text{erf}(\mu/\sqrt{2}\sigma)}{2} - \frac{\text{erf}((t_{p_a} - \mu)/\sqrt{2}\sigma)}{2}} \cdot \frac{1 - \frac{\text{erf}(\mu/\sqrt{2}\sigma)}{2} - \frac{\text{erf}((t_{p_{b_i}} + t_i - \mu)/\sqrt{2}\sigma)}{2}}{1 - \frac{\text{erf}(\mu/\sqrt{2}\sigma)}{2} - \frac{\text{erf}((t_{p_{b_i}} - \mu)/\sqrt{2}\sigma)}{2}} \quad (7)$$

4 Comprehensive Resource Management Algorithm

Suppose that there are m pieces together for a shared file. They comprise a group $P = \{p_1, p_2, \dots, p_m\}$. In a session whenever a peer receives any content sharing requests from others, it doesn't need to know the global network

topology information. Its decision for the shared pieces assignment among all applicants depends on its local knowledge as peer a in Figure 2. Let the peer that receives n content sharing requests be peer a . Those n peers form a group $B = \{b_1, b_2, \dots, b_n\}$. For a peer b_i , part of the resource file is stored in its local storage area. Thus, each peer b_i may have different content request upon the source peer a . Let the pieces of content requested by peer b_i constitute the group $P_{b_i} = \{p_{b_{i_1}}, p_{b_{i_2}}, \dots, p_{b_{i_k}}\}$ with $b_{i_1}, b_{i_2}, \dots, b_{i_k} \in \{1, 2, \dots, n\}$. A comprehensive resource management algorithm for file block assignment is shown in Figure 3.

```

while  $B_{ul_a} \neq 0$ 
   $b_i = Extract_{peer}(B)$ 
   $p_j = Extract_{block}(P_{b_i})$ 
   $B = min(B_{ul_a}, B_{dl_i})$ 
  Block  $P_j$  is sent to peer  $b_i$  from the source peer  $a$  through a channel with
  bandwidth  $B$ 
   $B_{ul_a} = B_{ul_a} - B$ 
end while

```

Fig. 3. The comprehensive resource management algorithm

Whenever a peer has part of content available and some requests from others, the proposed algorithm can be performed inside the peer. The resource assignment algorithm ends when all its outgoing channel bandwidth consumed. Two functions are conducted to select the right peer as the receiver and the block to be sent:

- $Extract_{peer}(B)$ is an instruction that will select a peer b_i from the group B and remove it. The selected peer has the maximum value of $P(B_i)$ among the group, which is defined in Equation 4. More specifically, the value of $P(B_i)$ can be calculated according to the employed model for the duration time of a peer as discussed in Section 3. Equation 5, 6 and 7 define the probability of a peer b_i to retrieve all needed data from the requested peer. If there is a tie, a peer has smaller f_i is selected.
- $Extract_{block}(P_{b_i})$ indicates which piece of content is chosen. After the decision of peer b_i to be the one served, the block among P_{b_i} that has been mostly requested in peer a is removed and assigned to p_j . In other words, compared with other blocks in the group P_{b_i} , p_j has the maximum number of peers that would like to copy it. If there is a tie, the block is arbitrarily selected.

In the proposed complete resource management algorithm, the peer that has the high probability to become a potential seed is the one to get service firstly, which is implemented by the message $Extract_{peer}(B)$. The maximum bandwidth for data transfer is $min(B_{ul_a}, B_{dl_i})$. The block of content to be copied first is the one most popularly requested. If there is still some abundant B_{ul_a} left, another loop of peer and block selection is executed.

5 Conclusion

In this paper, the resource availability issue has been addressed for a P2P network in terms of content sharing. When more peers containing the whole file, the P2P system is more stable in the sense that the data delivery traffic is evenly distributed and fault tolerance property is well achieved. The comprehensive resource management algorithm is proposed in order to maintain more seeds available. In the presented algorithm, the peer and the piece of content selection problem is solved. The peer with a higher probability to become a potential seed has the priority to be firstly served. The piece of content that has been mostly requested will be delivered first. We believe that the presented resource allocation algorithm can outperform randomized resource assignment techniques in most cases.

References

1. K. Aberer, M. Puceva, M. Hauswirth, and R. Schmidt. Improving data access in P2P systems. *IEEE Internet Computing*, 6(1):58–67, Jan.-Feb. 2002.
2. D. Barkai. Technologies for sharing and collaborating on the net. In *Proceedings of First International Conference on Peer-to-Peer Computing*, pages 13–28, Aug. 2001.
3. I. Clarke, S.G. Miller, T.W. Hong, O. Sandberg, and B. Wiley. Protecting free expression online with freenet. *IEEE Internet Computing*, 6(1):40–49, Jan.-Feb. 2002.
4. Eric W. Weisstein. “Erf.” From MathWorld-A Wolfram Web Resource. <http://mathworld.wolfram.com/Erf.html>.
5. A. Grimshaw, A. Ferrari, F. Knabe, and M. Humphrey. Wide area computing: resource sharing on a large scale. *IEEE Computer*, 32(5):29–37, May 1999.
6. Xiaohui Gu and K. Nahrstedt. A scalable QoS-aware service aggregation model for peer-to-peer computing grids. In *Proceedings of 11th IEEE International Symposium on High Performance Distributed Computing*, pages 73–82, July 2002.
7. K. Kant, R. Iyer, and V. Tewari. A framework for classifying peer-to-peer technologies. In *Proceedings of 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid*, pages 338–345, May 2002.
8. R. Matei, A. Iamnitchi, and P. Foster. Mapping the gnutella network. *IEEE Internet Computing*, 6(1):50–57, Jan.-Feb. 2002.
9. D.A. Menasce. Scalable P2P search. *IEEE Internet Computing*, 7(2):83–87, March-April 2003.
10. G. Pandurangan, P. Raghavan, and E. Upfal. Building low-diameter peer-to-peer networks. *IEEE Journal on Selected Areas in Communications*, 21(6):995–1002, Aug. 2003.
11. S. Waterhouse, D.M. Doolin, G. Kan, and A. Faybishenko. Distributed search in P2P networks. *IEEE Internet Computing*, 6(1):68–72, Jan.-Feb. 2002.