Policy-Driven Multi-File Distribution

Catherine Rosenberg
School of Electrical and
Computer Engineering
Purdue University
West Lafayette, IN 47907
Email: cath@ecn.purdue.edu

Pascal Pons
Department of
Computer Science
Ecole Normale Superieure
Paris, France
Email: pascal.pons@ens.fr

Dongyan Xu
Department of
Computer Sciences
Purdue University
West Lafayette, IN 47907
Email: dxu@cs.purdue.edu

***Abstract*— We propose to study the impact of a suite of policies on the performance of a multi-file distribution system that integrates CDN and P2P techniques. One of the policies is the peer contribution policy that decides the limited data rate and data volume to be contributed by each peer. The peer contribution policy is critical to maintaining the system's overall file distribution capacity without unfairly overloading the individual peers. In our previous work, we present an analytical framework for the modeling of a hybrid CDN-P2P architecture under a *file-specific* peer contribution policy. In this paper, we focus on a different scenario where *multiple* files are being distributed and the peer contribution policy is *file-independent*. We argue that a suite of policies need to be studied, in order to understand their impacts on the overall file distribution performance. The policies include: (1) file-independent peer contribution policy, (2) file request admission policy, (3) supplier selection policy, and (4) file replacement policy. We define a system model for the analysis of these policies. Based on the model, we also propose possible definitions of the policies.**

I. Introduction

Recent years have witnessed the increasing demand for large-volume data such as digital media and massive scientific data. The distribution of high-volume data poses new challenges and has led to cost-effective file distribution techniques. We have shown in [1] that the P2P (Peer-to-Peer) architecture, if properly jumpstarted by the CDN (Content Distribution Network) capacity, can dynamically generate and maintain aggregated file distribution capacity that satisfies subsequent file requests with low request rejection rate. In P2P data distribution, an individual peer offers limited storage space and out-bound data rate. For this reason, the contribution of each peer to the file distribution process needs to be carefully determined, in order to aggregate sufficient distribution capacity while keeping the contribution fair among the peers.

Our peer contribution policy in [1] is based on the volume of the file that is being distributed. Each peer is requested to transmit a total amount of data equal to r ($r \geq 1$) times the file size, regardless of the out-bound data rate it commits. An analysis is presented to capture the relation between the overall file distribution capacity and the individual peer contribution, demonstrating the impact of the latter on the former. However, the analysis in [1] assumes that the peer contribution policy is *file-specific*. In other words, after a peer receives a specific file f, it will fulfill its contribution commitment by re-distributing only the content of file f.

In this paper, we consider a peer contribution policy that is *file-independent*: Multiple files are being distributed in the system; and each peer only maintains *one* contribution contract based on the total volume of data it has received, rather than multiple contribution contracts - each for a different file. Interestingly, under this new scenario, a suite of new problems arise. Each problem is associated with a policy which may affect the overall file distribution performance of the system. The first policy is the *file request admission policy*: with multiple files being distributed, the admission of a file request from a peer may have to be decided by the peer's contribution fulfillment status. The second policy is the *supplying peer selection policy*: with the supplying peers of a file making varying progress in their contract fulfillment, the choice of supplying peers for each file request may affect the overall distribution capacity of different files. The last policy is the *file replacement policy*: with limited storage space for P2P distribution, each peer needs to carefully decide which files to retain and which files to discard, when the P2P storage space is full. To further complicate the analysis, these policies are *not* orthogonal. Instead, they need to be designed and analyzed in an integrated fashion.

The purpose of this paper is to motivate an in-depth study of these new policies (including the new file-independent peer contribution policy), and capture their impacts on the file distribution performance. We present a comprehensive system model as the basis for further investigation. We also propose candidate (not necessarily optimal) policies for multi-file distribution and describe our policy-making principles. Results from the proposed study will be especially useful in the planning and dimensioning of systems involving concurrent and continuous release of files, such as in the distribution of news and movies.

The rest of this paper is organized as follows: Section II presents a model for the multi-file distribution system. Section III proposes a suite of policies for multi-file distribution. Section IV discusses related work. Finally, Section V concludes this paper and suggests open research problems.

II. System Architecture and Model

We assume the same hybrid system architecture as in [1], with one CDN server (or *"server"* for the rest of the paper) and a peer community it serves. The server releases files for distribution on a continuous basis. It is also the manager of the file distribution system, accepting peer requests and making decisions on file request admission, supplying peer selection, and file replacement in peers, based on the suite of file distribution policies to be described in Section III. For analysis convenience, we assume *centralized* enforcement of the policies, which does not necessarily reflect its real-world implementation. The peers receive files from the system. On the other hand, they will be required to re-distribute the files they have received to other peers, based on a *peer contribution policy*. For modeling simplicity, we focus on file distribution in a local region, assuming that the intermediate network connecting the server and the peers is not the bottleneck.

A. The Server

Modeled as one logical entity, the server is the source of all files distributed in the system. For modeling convenience, we assume that it has a global view of the system, with complete information about each peer and each file. In particular, for each peer, it keeps track of the files that the peer is keeping for re-distribution, as well as the peer's contribution contract fulfillment status. For each file request, the server performs admission control to determine if the request can be admitted. If the request is admitted, the server will further decide if the request will be served by the server itself or by peers. If the latter is the case, the server will select a set of peers as suppliers. Furthermore, the server also manages the limited P2P storage space set aside in each peer, by deciding which files should be kept by the peer among the files that the peer has received. To jumpstart the P2P distribution capacity, the server itself allocates a capacity of C_{server} to the system, which can be considered as the out-bound bandwidth for file distribution.

B. The Peers

The community of registered peers is a finite set $\mathcal{P}$ with N peers. The attributes of each peer $p \in \mathcal{P}$ include: (1) the downloading bandwidth $C_{in}(p)$; (2) the committed out-bound data rate $C_{out}(p)$ for file re-distribution; (3) the storage space set aside for the system $Space(p)$ - this space is used for keeping a *limited* number of files that the peer has received from the system. When the storage space is full, the file replacement decision will be made by the server; (4) the debt $Debt(p,t)$ (in data volume) to the system at time t - a peer with zero debt does not have to contribute; while a peer with a positive debt is called an "active peer" and is expected to distribute $Debt(p,t)$ amount of data to other peers. The initial debt of all peers is zero ($Debt(p,0) = 0$).

C. The Files

At a given time t, the set of files being distributed in the system is $\mathcal{F}(t)$. Over the time, the server releases new files and removes old files from the system. The release of new files can either be deterministic (for example, a set of new files every day) or be a stochastic process. Each file $f \in \mathcal{F}$ has the following attributes: (1) the size of f $Size(f)$; (2) the initial release time of f by the server $T_f \geq 0$; (3) the removal time of f from the server $\tilde{T_f}$ - note that after $\tilde{T_f}$, f can still be distributed by the peers in the system; and (4) the mean request rate $\lambda_f(t)$ for file f at time t.

D. The Peer Contribution Contract

Each peer in the system is required to fulfill a dynamic and file-independent contribution contract, enforced by the server. When a peer p completes the downloading of a file f, its debt $Debt(p,t)$ is increased by $\beta(f,t) \times Size(f)$. $\beta(f,t)$ is a time-varying *contribution factor*. When a peer p provides K amount of data to another peer, the debt of p is decreased by K. The debt $Debt(p,t)$ of a peer may be bounded by a maximum debt $MaxDebt(p)$ such that: $\forall p \in \mathcal{P}, 0 \leq Debt(p,t) \leq MaxDebt(p)$. A peer with zero debt will be freed from the re-distribution duties.

III. Policies for Multi-File Distribution

With the introduction of file-independent peer contribution contract, it becomes necessary to investigate a number of policy issues which are critical to the overall file distribution performance. To the best of our knowledge, there has been no systematic study on these policies and their impacts on the file distribution process. In this section, we first identify the suite of policies for multi-file distribution. We then define candidate policies and discuss their design principles.

A. A Suite of Policies

We propose to study the following multi-file distribution policies:

- *The file request admission policy* This policy is expected to differentiate requesting peers based on the status of their contribution contract fulfillment status. Consequently, it will have impact on the aggregated distribution capacity for different files in $\mathcal{F}(t)$, by giving preference to peers that will help to increase the capacity for highly demanded files.
- *The supplier selection policy* When a request is admitted, the server has to decide which supplying peers will be selected to serve this request. The chosen peers must have a copy of the requested file and be available at that time. If there are more qualified supplying peers than needed, a selection must be made according to the peers' current debts, other files stored, and committed out-bound data rate $C_{out}(p)$. Intuitively, it should be avoided that a peer that has a highly requested file be selected to serve the request for a much less popular file. The policy should also avoid creating peers with excessively high debt.
- *The file replacement policy* Due to the limited $Space(p)$ of each peer, a file replacement policy is needed to decide which file should be discarded by a peer when the storage space is full. The policy will determine how many peers system-wide should store a specific file: If too many peers keep this file, it will lead to a waste of space. However, if there are not enough peers storing the file, low file

request admission rate will occur, even if the distribution capacity (or, the overall debt) abounds among the peers.
- *The peer contribution policy* A good contribution policy should avoid an infinite growth of the total debt. Instead, it should maintain the debt of each peer at an appropriate level, in order to ensure that there exist enough active peers in the system at any time. It is possible to control the total debt of peers by imposing appropriate contribution factor $\beta(f,t)$ at different times. The contribution factor may be bounded such that $\beta_{min} \leq \beta(f,t) \leq \beta_{max}$.

B. Useful Variables and Parameters

Before proposing our policies for multi-file distribution, we first define a number of useful variables and parameters, in order to characterize the supply and demand of files.

- C_{in}: the average downloading bandwidth of the peers. $C_{in} = \frac{1}{N} \sum\limits_{p \in \mathcal{P}} C_{in}(p)$.
- C_{out}: the average committed out-bound data rate of the peers. $C_{out} = \frac{1}{N} \sum\limits_{p \in \mathcal{P}} C_{out}(p)$.
- $\alpha(p,t)$ and $\alpha(f,p,t)$: the participation probability of peer p. We define the participation probability $\alpha(p,t)$ of p as the probability that p is busy serving a request at time t. Thus the mean bandwidth provided by p is $\alpha(p,t)C_{out}(p)$. Now we consider a moment t when p is serving: we define the participation probability of p for file f as $\alpha(f,p,t)$: the probability that p is re-distributing f at this moment.
- $C_{requested}(f,t)$: the total requested bandwidth for file f at time t. Since the request rate for file f is $\lambda_f(t)$, we have $C_{requested}(f,t) = \lambda_f(t) \times \frac{Size(f)}{C_{in}} \times C_{in}$, in which $\frac{Size(f)}{C_{in}}$ is the expected downloading time of file f. We thus have $C_{requested}(f,t) = \lambda_f(t) \times Size(f)$.
- $Contrib(f,t)$: the expected bandwidth contribution from peers for the distribution of f. For each peer, we expect that its contribution is fairly distributed among the files it currently keeps. Therefore, the expected participation ratio of peer p for file f is $\frac{Size(f)}{Space(p)}$; and the expected contribution is $Contrib(f,t) = \sum\limits_{p \in \mathcal{P}} \delta(f,p,t) \frac{Size(f)}{Space(p)} C_{out}(p)$, where $\delta(f,p,t) = 1$ if p has f at t; and $\delta(f,p,t) = 0$ otherwise. The actual total bandwidth contribution from peers for f can be greater than the expected bandwidth contribution at a given time.
- $Ch(f,t)$: the *demand-supply ratio* of file f at t. In the file distribution system, it is desirable that the demand - the total requested bandwidth for file f ($C_{requested}(f,t)$) - be met by the supply - the expected bandwidth contribution from peers for f ($Contrib(f,t)$). Therefore, the demand-supply ratio of file f is defined as $Ch(f,t) = \frac{C_{requested}(f,t)}{Contrib(f,t)}$. If the value of $Ch(f,t)$ for file f is too large, it means that the demand cannot be met by the supply; and the peers that have f will have to contribute more to the distribution of f. To be fair among all the files, our policies should try to narrow the difference in $Ch(f,t)$ among these files.

C. Proposed Policies

In this section, we propose our definitions of the policies. For each policy, we will always begin by presenting a simple “benchmark” policy and then describe our policy. The benchmark policies will be used to compare with our policies so that the performance improvement achieved by the latter may be demonstrated. The key principle behind our policies is to *balance* the demand-supply ratio among all the files being distributed.

1) The Supplier Selection Policy: When a peer p submits a file request, the system tries to provide a file downloading rate that is equal to p’s in-bound bandwidth $C_{in}(p)$. If this rate cannot be reached, the system provides the highest possible rate. If no supplying peers (peer having f) are available and the server does not have free capacity, the request will be rejected.

Benchmark policy: In the simple benchmark policy, the system always first selects the available supplying peers of f that have the *highest debt*. If there are not enough peers to provide $C_{in}(p)$, the server may provide the remaining bandwidth if available.

Our policy: The benchmark policy is very simple. However it is already a good policy. By selecting the peers with the greatest debt, the policy distributes debts more evenly among the peers and therefore increases the percentage of active peers (namely, peers with positive debt) in the system. Our policy is based on a similar principle, but has the nice property that the participation probability $\alpha(f,p)$ can be estimated. When selecting the supplying peers of file f, our policy randomly chooses enough peers among the available peers, with a weight assigned to each peer p as $\frac{C_{out}(p)}{Space(p)} Debt(p,t)$. With some approximation, we can show that in this case $\alpha(p,t) = \min(1, \frac{Debt(p,t)}{MeanDebt(t)} \sum\limits_{g \in \mathcal{F}} (\delta(g,p) Ch(g,t) \frac{size(g)}{Space(p)})$ and $\alpha(f,p,t) = \frac{Ch(f,t) size(f)}{\sum\limits_{g \in \mathcal{F}} \delta(g,p) Ch(g,t) size(g)}$. Especially, if the demand-supply ratios of all the files kept by p are equal, we will simply have $\alpha(f,p,t) = \frac{size(f)}{Space(p)}$.

2) The File Replacement Policy: When a peer has received a requested file, if there is enough space in its P2P storage, the file will always be kept in the P2P storage for re-distribution. Otherwise, the peer may save the file by replacing other files. This choice is guided by the file replacement policy.

Benchmark policy: The simplest replacement policy is to always save the new file received. If there is no sufficient space, a randomly selected file will be deleted. For simplicity, we assume that all files are of equal size for the rest of this paper.

Our policy: The replacement policy is strongly related to the selection policy, both influencing the number of peers that supply each file in the system. The replacement policy helps to control the distribution and placement of files among the peers. A good file placement will in turn improve the effectiveness of the selection policy. Since our goal is to narrow the difference in demand-supply ratio among all files, we propose a replacement policy that will make the demand-supply ratio of all files converge to the same value.

There are two events that can change the value of $Ch(f,t)$: When a peer deletes f, $Ch(f,t)$ increases. When a peer saves f, $Ch(f,t)$ decreases. To make the values of $Ch(f,t)$ converge, the policy will replace a file with a lower $Ch(f,t)$ with a file having a higher $Ch(f,t)$. Therefore, when a peer has received file f, the peer will save f only if it can find another file g with a lower $Ch(g,t)$. In our policy, let g be the file saved by the peer that has the lowest $Ch(g,t)$. If $Ch(f,t) > Ch(g,t)$, the system will replace g with f. Otherwise, f will not be saved.

However, this policy may completely delete an unpopular file from all peers in the system. In this case, when a new request for this file arrives, there may not be any available peers to serve this request. To avoid the complete deletion of an unpopular file that is still being requested, we will impose a number N_{min}, which is the minimum number of peers that store a file f in the system. Let $N_f(t)$ be the number of peers having f at time t. When $N_f(t) \leq N_{min}$, we stop the replacement policy from deleting f from a peer. However, in order to determine the time when a file really needs to be deleted from the system, the policy stipulates that the system only keeps the set of files $\mathcal{F}_{saved} = \{f \in \mathcal{F} | \lambda_f(t) > \lambda_{min}\}$.

3) The Peer Contribution Policy: In the file-independent peer contribution policy, the debt of a peer will be bounded: $0 \leq Debt(p,t) \leq MaxDebt(p)$. A peer that reaches zero debt will not have to contribute. When a peer reaches the upper bound $MaxDebt(p)$, it will still be able to request files but its debt will not increase. For example, we may set $MaxDebt(p) = 4$ times the size of a file. The key factor in the contribution policy is $\beta(f,t)$, the contribution factor for the distribution of f at time t. The value of $\beta(f,t)$ is also bounded: $\beta_{min} \leq \beta(f,t) \leq \beta_{max}$. For example, we may set $\beta_{min} = \frac{1}{2}$ and $\beta_{max} = 2$.

Benchmark policy: The simplest contribution policy is to require that each peer gives back exactly the same amount of data that it receives. Therefore, the contribution factor $\beta(f,t)$ is always equal to 1. The net growth of the total debt of all the peers ($D(t)$) is due to the contribution of the server. On the other hand, the net loss of contribution is due to the upper bound of debt $MaxDebt(p)$.

Our policy: The mean debt of peers $\frac{1}{N} \sum\limits_{p \in \mathcal{P}} Debt(p,t)$ is an important parameter that determines the peer distribution capacity in the system. The higher the mean debt, the greater the capacity. However, increase in mean debt will not lead to infinite increase in the overall file distribution capacity. Therefore, an appropriate target mean debt needs to be determined. We assume that an ideal mean debt of a peer $IdealDebt$ is set by the policy. If the actual mean debt is less than $IdealDebt$, $\beta(f,t)$ will be dynamically set as greater than 1.0. If the actual mean debt is greater than $IdealDebt$, $\beta(f,t)$ will be reduced to less than 1.0. More specifically, we may define:

$$\beta(f,t) = 1 + \gamma(1 - \frac{\frac{1}{N} \sum\limits_{p \in \mathcal{P}} Debt(p,t)}{IdealDebt}) \qquad (1)$$

γ is a bounding factor. For example, we may set $\gamma = 5$ to ensure that the mean debt of a peer will never be greater than $1.2 \times IdealDebt$

At the beginning of the distribution process of file f, it is useful to make the peers contribute more to the distribution of f. By doing this, we can increase the ratio of active peers for f and build up its distribution capacity. Moreover, if the contribution factor $\beta(f,t)$ is large, some peers may prefer to wait until the file becomes less "expensive", thus lowering the request rate of file f. We will compute $\beta(f,t)$ according to the demand-supply ratio $Ch(f,t)$. It is expected that $\beta(f,t) = 1$ when $Ch(f,t) = 1$. However, we cannot always expect $\beta(f,t) = Ch(f,t)$ at any time t. In fact, $Ch(f,t)$ tends to be too high for $\beta(f,t)$ to match at the beginning. Rather, $\beta(f,t)$ can be defined as an affine transformation of $ch(f,t)$, such that $\beta(f,t) = 1$ when $Ch(f,t) = 1$; and that $\beta(f,t) = A$ when $Ch(f,t) = M$ $(1 < A < M)$ - both A and M are tunable in the policy. We can then derive an expression of $\beta(f,t)$ by revising (1) as:

$$\beta(f,t) = (1 + (ch(f,t) - 1)\frac{A-1}{M-1}) \times$$

$$(1 + \gamma(1 - \frac{\frac{1}{N} \sum\limits_{p \in \mathcal{P}} Debt(p,t)}{IdealDebt}))$$

Our preliminary simulation study shows that the proposed suite of policies achieve significant improvement in file distribution performance, compared with the benchmark policies. The performance metrics include file request admission rate, average file downloading time, file distribution time (i.e. total time needed to distribute a file to all interested peers), and contribution fairness among peers.

IV. Related Work

An integrated and measurement-based study is presented in [2] on Internet content delivery systems, including HTTP web traffic, CDN (Akamai) and P2P (Gnutella and Kazaa). The study verifies the increasing popularity of P2P-based content delivery and characterizes the behavior of different content delivery systems. An emerging content distribution scheme is based on the integration of CDN and P2P architectures. Such a hybrid architecture has been shown to be highly cost-effective [1] [3].

Analytical models have been proposed to study the dynamics of P2P systems. In [4], a P2P file sharing system is modeled as a multi-class closed queuing network. This allows for the analysis of system throughput dynamics under various configurations of the peer community. Different from the file distribution capacity defined in our model, the throughput analysis in [4] does not consider the limited peer bandwidth contribution. Also, it assumes one supplier for each file request.

Incentive-based mechanisms aim at encouraging peers to contribute to the P2P community [5] [6] [7] [8]. Also related is the free riding problem: [9] and [10] investigate the problem through measurement study and through game-theoretic analysis, respectively. Both [9] and [10] advocate the use of payment mechanisms in order to motivate the peers with incentives to contribute to the system. Instead of an abstract payment model, our work adopts a simpler model that directly associates the volume of data received by a peer with the volume of data the peer is supposed to re-distribute.

Reputation models have also been proposed for P2P systems [11] [12]. A reputation-based admission control algorithm is proposed in [12]: The reputation of each peer is based on its past contributions and is computed using a distributed eigenvector method. A peer requesting a specific service would have to acquire a certain level of reputation, i.e., to have made a certain amount of contributions. While reputation-based methods motivate peers to behave properly, it is not clear if the methods also lead to optimal P2P service capacity growth and distribution.

V. Conclusion and Future Work

We have proposed a framework for the study of different file distribution policies in a hybrid CDN-P2P architecture. We argue that in the presence of multiple files in the system, the distribution processes of different files will interfere with each other, and a file-independent peer distribution policy needs to be carefully designed to achieve optimal generation and allocation of file distribution capacity. Furthermore, the policies of file request admission, supplying peer selection, and file replacement all have impact on the overall file distribution performance. Especially, they need to be aware of the dynamic demand-supply status of different files in the system.

While making a case for policy-driven multi-file distribution, this paper leads to research problems rather than solutions. A detailed yet tractable analysis is needed to model the system dynamics under the file distribution policies. More specifically, the analysis is expected to show that the policies achieve convergence of demand-supply ratios among all files. Another challenge is to compare the proposed model with the incentive-based and reputation-based models, so that a uniform and comprehensive framework for P2P system capacity planning and distribution can be established. Finally, efforts are needed to design and analyze protocols for enforcing the proposed policies in a fully distributed (rather than centralized) fashion.

Acknowledgment

The authors would like to thank the anonymous reviewers for their helpful comments. This work is supported by a grant from the e-Enterprise Center at Discovery Park, Purdue University.

References

[1] D. Xu, H.K. Chai, C. Rosenberg, S. Kulkarni. *Analysis of a Hybrid Architecture for Cost-Effective Streaming Media Distribution*, SPIE/ACM Conf. on Multimedia Computing and Networking (MMCN'03), San Jose, CA, Jan. 2003.

[2] S. Saroiu, K. Gummadi, R. Dunn, S. Gribble, H. Levy. *An Analysis of Internet Content Delivery Systems*, USENIX Symposium on Operating Systems Design and Implementation (OSDI'02), Boston, MA, Dec. 2002.

[3] L. Guo, S. Chen, S. Ren, X. Chen, S. Jiang. *PROP: a Scalable and Reliable P2P Assisted Proxy Streaming System*, 24th International Conference on Distributed Computing Systems (ICDCS'04), Tokyo, Japan, Mar. 2004.

[4] Z. Ge, D. Figueiredo, S. Jaiswal, J. Kurose, D. Towsley. *Modeling Peer-to-Peer File Sharing Systems*, IEEE INFOCOM'03, San Francisco, CA, Mar. 2003.

[5] Q. Sun, H. Garcia-Molina. *SLIC: a Selfish Link-Based Incentive Mechanism for Unstructured Peer-to-Peer Networks*, 24th International Conference on Distributed Computing Systems (ICDCS'04), Tokyo, Japan, Mar. 2004.

[6] R.T.B. Ma, S.C.M. Lee, J.C.S. Lui, D.K.Y. Yau. *An Incentive Mechanism for P2P Networks*, 24th International Conference on Distributed Computing Systems (ICDCS'04), Tokyo, Japan, Mar. 2004.

[7] K. Anagnostakis, M. Greenwald. *Exchange-Based Incentive Mechanisms for Peer-to-Peer File Sharing*, 24th International Conference on Distributed Computing Systems (ICDCS'04), Tokyo, Japan, Mar. 2004.

[8] W. Wang, B. Li. *To Play or to Control: A Game-based Control-theoretic Approach to Peer-to-Peer Incentive Engineering*, IEEE/IFIP IWQoS'03, Monterey, CA, Jun. 2003.

[9] E. Adar, B. Huberman. *Free Riding on Gnutella*, First Monday, 5(10), 2000.

[10] P. Golle, K. Leyton-Brown, I. Mironov. *Incentives for Sharing in Peer-to-Peer Networks*, Second Workshop on Electronic Commerce (WELCOM'01), Heidelberg, Germany, 2001.

[11] S. Kamvar, M. Schlosser, H. Garcia-Molina. *The EigenTrust Algorithm for Reputation Management in P2P Networks*, Twelfth International World Wide Web Conference (WWW'03), Budapest, Hungary, May 2003.

[12] H.T. Kung, C. Wu. *Differentiated Admission for Peer-to-Peer Systems: Incentivizing Peers to Contribute Their Resources*, Workshop on Economics of Peer-to-Peer Systems, Berkeley, CA, Jun. 2003.