A Game Theoretic Approach to Provide Incentive and Service Differentiation in P2P Networks

Richard T. B. Ma,   Sam C. M. Lee,
John C. S. Lui[*]
Dept of Computer Science & Engineering, The
Chinese University of Hong Kong

{*tbma,cmlee,cslui*}@*cse.cuhk.edu.hk*

David K. Y. Yau[†]
Dept of Computer Science
Purdue University
West Lafayette, IN 47907

yau@cs.purdue.edu

ABSTRACT

Traditional peer-to-peer (P2P) networks do not provide service differentiation and incentive for users. Consequently, users can obtain services without themselves contributing any information or service to a P2P community. This leads to the "free-riding" and "tragedy of the commons" problems, in which the majority of information requests are directed towards a small number of P2P nodes willing to share their resources. The objective of this work is to enable service differentiation in a P2P network based on the amount of services each node has provided to its community, thereby encouraging all network nodes to share resources. We first introduce a resource distribution mechanism between all information sharing nodes. The mechanism is driven by a distributed algorithm which has linear time complexity and guarantees *Pareto-optimal* resource allocation. Besides giving incentive, the mechanism distributes resources in a way that increases the aggregate utility of the whole network. Second, we model the whole resource request and distribution process as a competition game between the competing nodes. We show that this game has a Nash equilibrium and is collusion-proof. To realize the game, we propose a protocol in which all competing nodes interact with the information providing node to reach Nash equilibrium in a dynamic and efficient manner. Experimental results are reported to illustrate that the protocol achieves its service differentiation objective and can induce productive information sharing by rational network nodes. Finally, we show that our protocol can properly adapt to different node arrival and departure events, and to different forms of network congestion.

Categories and Subject Descriptors

C.2 [**Computer-Communication Networks**]: Distributed Systems; C.4 [**Performance of Systems**]: Design studies

General Terms

Algorithm, Design, Economics

1. INTRODUCTION

Recently, there has been tremendous interest in P2P networks. As evidenced by traffic measurement data of ISPs, a large percentage of today's Internet traffic is due to P2P applications [2], which aim to exploit the cooperative paradigm of information exchange to greatly increase the accessibility of information to a large number of users.

Unlike traditional client-server networking, P2P networks allow individual users (or *nodes*) to play the roles of both client and server at the same time. As such, nodes can help each other in file searching [1], file lookup [14, 15, 18, 20], and anonymous information transfer [6]. For file searching, P2P networks have evolved from a centralized file/directory lookup approach (e.g., Napster) to a distributed object query approach (e.g., Gnutella). Whereas distributed object queries can be effected by some form of controlled flooding, the new generation P2P networks (e.g., Chord and CAN) use the method of *consistent hashing* to increase the file lookup efficiency.

Besides such efforts to improve the efficiency of file lookup and other operations in a P2P network, some fundamental and challenging issues remain unanswered about the basic cooperative paradigm of information exchange. *Free-riding* and the *tragedy of the commons* are two such problems. In [3], the authors report that nearly 70% of P2P users do not share any file in a P2P community. Instead, these users simply "free ride" on other users who do share. Since the users who are willing to share or provide services to others are few, nearly 50% of all file searching responses come from the top 1% of information sharing nodes. Therefore, nodes that share information and resources are prone to congestion, leading to the tragedy of the commons problem [9].

In this paper, we propose a protocol to provide service differentiation in a P2P network based on the contribution levels of individual nodes. Specifically, we focus on a *file transfer* service because the amount of data transferred per unit time is much higher than that of object lookup/query. Roughly speaking, a node which shares popular files and

provides more service (e.g., via file upload) to the P2P community will earn a higher contribution level. As a result, when this node later asks for a file transfer, it will be granted a higher utility than other competing nodes which have lower contribution levels. We address the challenges of incorporating such *incentive-compatible* resource distribution in the file transfer process such that we can: (1) provide proper service differentiation to network users, (2) encourage nodes to share information or services with their peers, and (3) maximize the social welfare [16] or the aggregate utility of the users. It is important to point out that our incentive protocol can be incorporated into various P2P systems that use either the distributed query (e.g., Gnutella) or the consistent hashing approach (e.g., Chord or CAN). In addition, our protocol has the following desirable properties:

1. *Fairness:* Nodes which have contributed more to the P2P network should gain more resources and achieve higher utilities in the resource sharing.

2. *Avoidance of resource wastage:* The mechanism will not assign more resources to a node than it can consume. In case there is congestion on the path of communication, the mechanism can adapt to the congestion level and redistribute resources accordingly.

3. *Adaptability and scalability:* The mechanism can adapt to conditions such as dynamic node join/leave. Since the mechanism runs at each participating node, its performance is scalable as the size of the P2P network increases.

4. *Maximization of social utility:* Under certain circumstances, our mechanism does not only maximize the utilities of individual users, but also achieves high aggregate utility for the set of all users.

As we will show, the proposed mechanism requires requesting users to bid for resources, thereby creating a *dynamic competition game*. To ensure that every node in the P2P network will follow the mechanism honestly, the created dynamic game should be *strategy-proof* and *collusion-proof*. The first property implies that following the proposed mechanism is the *best* strategy for each user in the network. The second property implies that users cannot gain extra resources by cooperatively deceiving the system.

1.1   Related work

We now briefly present some related work. In [8], the authors propose an incentive mechanism for *centralized* P2P systems like Napster. Our work, on the other hand, can be applied to both centralized and distributed P2P networks. Zhong *et al.* [21] discuss the shortcomings of *micro-payment* and reputation systems. They propose a cheat-proof, credit-based mechanism for mobile ad-hoc networks. However, issues of providing incentive and service differentiation are not addressed in their work. In [7], the authors discuss the economic behavior of P2P storage networks only. In [19], the authors model P2P networks as a Cournot Oligopoly game and give an elegant control-theoretic solution focusing on a *global storage system* only. In contrast, our work addresses file transfer and bandwidth allocation in a P2P network, and we use the approach of algorithmic *mechanism design* [12, 13, 17] to design a desirable competition game for our system.

1.2   Paper organization

The balance of the paper is organized as follows. In Section 2, we give a general overview of how the node providing information interacts with the nodes seeking the information. In Section 3, we present our resource distribution mechanism and its properties. In Section 4, we present our dynamic game model and show how it can be applied to a P2P network. In Section 5, we present experimental results that illustrate the performance and effectiveness of the proposed mechanism and competition game. Section 6 concludes.

2.   INCENTIVE P2P SYSTEM OVERVIEW

In this section, we provide an overview of our incentive P2P system. In particular, we illustrate the interactions between different nodes during the file transfer process. In later sections, we will formally present the development of the resource distribution mechanism and its performance properties.

Similar to other P2P systems, each node in our incentive P2P network can play the roles of both server and client at the same time. During a file transfer, the node which provides the service (e.g., uploading files to other nodes) is called the *source node*, denoted as $\mathcal{N}_s$. Nodes which request file download from $\mathcal{N}_s$ are called the *competing nodes*, and are denoted as $\mathcal{N}_1, \mathcal{N}_2, \cdots, \mathcal{N}_N$, where N is the number of competing nodes. Each node in an incentive P2P network has a contribution value, which indicates how much service the node has provided to the whole P2P community. In order to maintain these values securely, there is an entity called the *auditing authority*, denoted as $\mathcal{A}$. One should view the auditing authority as a *distributed infrastructure*. For the implementation issues of $\mathcal{A}$, please refer to [10].
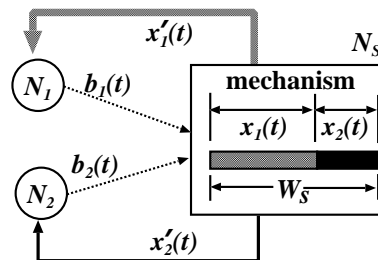

Figure 1: Illustrating two competing nodes and a source node.

Figure 1 illustrates a scenario where there are two competing nodes $\mathcal{N}_1$ and $\mathcal{N}_2$, which request file download service from the source node $\mathcal{N}_s$. The source node has an upload bandwidth resource of $\mathcal{W}_s$ (in bits/s). From time to time, these competing nodes send messages $b_1(t)$ and $b_2(t)$ (in bits/s) to $\mathcal{N}_s$, telling how much transfer bandwidth they desire. Upon receiving these messages, $\mathcal{N}_s$ will use a resource distribution mechanism (to be presented in Section 3) to distribute its bandwidth resource $\mathcal{W}_s$ based on the values of $b_1(t)$, $b_2(t)$, as well as the requesting nodes' contribution values denoted by $C_1(t)$ and $C_2(t)$, respectively. As a result, $\mathcal{N}_s$ sends the file data to $\mathcal{N}_1$ and $\mathcal{N}_2$ with bandwidth $x_1(t)$ and $x_2(t)$, respectively. However, it is possible that there is network congestion along the communication path between $\mathcal{N}_s$ to $\mathcal{N}_1$ (or $\mathcal{N}_2$). Therefore, packets may be lost

and the actual received bandwidth at node $\mathcal{N}_1$ and $\mathcal{N}_2$ are $x'_1(t) \leq x_1(t)$ and $x'_2(t) \leq x_2(t)$, respectively.

The auditing authority $\mathcal{A}$ in an incentive P2P network is a distributed database which serves two important functions. First, the auditing authority $\mathcal{A}$ will reply with the contribution value of any node upon request. For example, the source node $\mathcal{N}_s$ needs to know the contribution values of its competing nodes so as to distribute its resources accordingly. Second, the auditing authority $\mathcal{A}$ maintains or increments the contribution value of a node, say $\mathcal{N}_s$, when $\mathcal{N}_s$ presents the *evidence* that it has performed some service for other nodes. As mentioned before, a source node will receive messages (e.g., b_i) from the competing nodes and these can be used as evidence for contribution update. Figure 2 illustrates the two functions of $\mathcal{A}$. Again, for the implementation of $\mathcal{A}$, as well as related issues on security and collusion, please refer to [10].
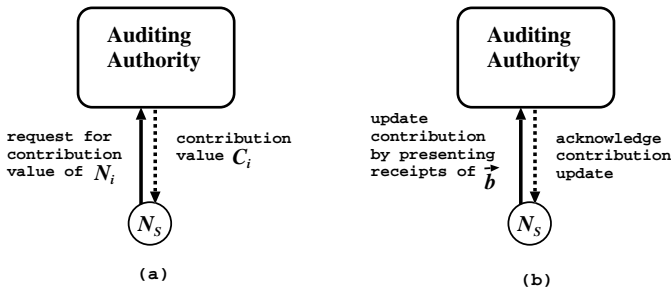

Figure 2: Illustration of functionalities of auditing authority: (a) reply contribution value upon request; (b) update contribution value upon request.

The message $b_i(t)$ plays two important roles. First, it can be regarded as a *bandwidth bidding* message from the perspective of the competing node $\mathcal{N}_i$. Another usage of $b_i(t)$ is that it is a *confirmation* to the source node $\mathcal{N}_s$ that $\mathcal{N}_i$ has received a certain amount of service (measured in unit of bits/s). This kind of message helps the source node to determine the proper bandwidth assignment. If a competing node is inactive or failed, the source node will assume that the competing node cannot receive any data. Therefore, it will not send any more packet to the competing node. The source node, on the other hand, can *adjust* the bandwidth resource assignment whenever it receives a bidding message. The justifications for this adjustment are: (1) a newly arriving competing node may request $\mathcal{N}_s$ for a new file download, (2) an existing competing node finishes its file transfer service, and (3) due to network congestion situation, a competing node replies with different values of bidding messages throughout the file download session. To efficiently utilize the bandwidth resource $\mathcal{W}_s$ and to improve the rate of contribution increase for $\mathcal{N}_s$, the source node needs to adjust the bandwidth distribution among competing nodes.

Figure 3 illustrates the interactions between the competing nodes and the source node $\mathcal{N}_s$. At time t_0, the competing node $\mathcal{N}_i$ requests the transfer of a large file F_i and sends a bidding message $b_i(t_0)$ to $\mathcal{N}_s$. After verifying the identity and contribution level of $\mathcal{N}_i$, $\mathcal{N}_s$ uses the resource distribution mechanism to determine the sending bandwidth $x_i(t_0)$, and delivers some data packets of F_i to $\mathcal{N}_i$ based on this rate allocation. After receiving these data packets, $\mathcal{N}_i$ sends another bidding/receipt $b_i(t_1)$ at time t_1. $\mathcal{N}_s$ then determines
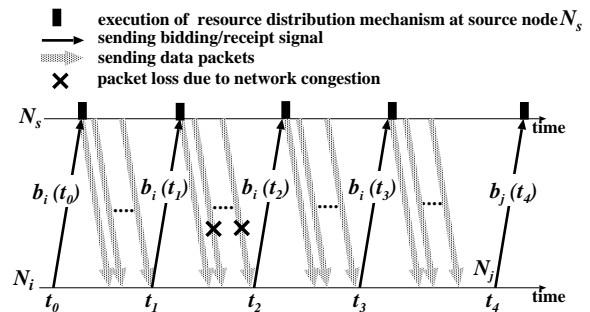

Figure 3: Interaction between competing nodes and a source node.

the new resource allocation and sends some additional data packets of file F_i based on $x_i(t_1)$. Note that at this round of the data delivery, some data packets are lost due to network congestion. Therefore, $\mathcal{N}_i$ sends a bidding/receipt $b_i(t_2)$ to $\mathcal{N}_s$ at time t_2, with $b_i(t_2) < b_i(t_3)$. The source node $\mathcal{N}_s$ adjusts the resource allocation and delivers additional data packets of file F_i to $\mathcal{N}_i$ at a lower rate. At time t_4, a new competing node $\mathcal{N}_j$ requests a transfer of the file F_j from $\mathcal{N}_s$ and sends its bidding message $b_j(t_4)$. $\mathcal{N}_s$ adjusts the resource allocation based on the latest biddings of these two competing nodes $\mathcal{N}_i$ and $\mathcal{N}_j$.

3. RESOURCE DISTRIBUTION MECHANISM

In this section, we discuss how the source node, say $\mathcal{N}_s$, uses the mechanism to distribute its bandwidth resource $\mathcal{W}_s$ (in bits/s) among all its competing nodes $\mathcal{N}_1, \ldots, \mathcal{N}_N$. For ease of presentation, we start with some simple mechanisms and will discuss their shortcomings. Then we introduce some more sophisticated features so as to provide service differentiation and incentive.

Even sharing mechanism (ESM): The first mechanism is to *evenly* divide the resource $\mathcal{W}_s$ among all competing nodes. When there are N competing nodes requesting file download, $\mathcal{N}_s$ transmits a file to a competing node $\mathcal{N}_i$ with an assigned bandwidth x_i:

$$x_i = \frac{\mathcal{W}_s}{N} \text{ for } i = 1, \ldots, N. \tag{1}$$

Although this mechanism seems fair in distributing the resource, there are some inherent problems. First, the bandwidth resource wastage may be significant. The wastage can occur in at least two forms: (1) if the connection between $\mathcal{N}_s$ and a competing node is congested, the assigned bandwidth is not fully utilized, and (2) the physical download bandwidth of a competing node may be less than the assigned bandwidth of $\mathcal{W}_s/N$; hence, the source node $\mathcal{N}_s$ cannot deliver information at that rate. Note that resource wastage also implies that $\mathcal{N}_s$ contributes some service to the community. However, the amount of work may not be counted toward its contribution. Another problem of this type of mechanism is that it provides no service differentiation among the competing nodes. Therefore, rational users have no incentive to share their information or service, resulting in the tragedy of the commons.

Resource bidding mechanism (RBM): The aim of this mechanism is to overcome the resource wastage problem

mentioned above. Under this mechanism, every competing node is required to send a bidding message periodically to $\mathcal{N}_s$. Let $b_i(t)$ be the bidding message from the competing node $\mathcal{N}_i$ at time t indicating the *maximum* bandwidth (in bits/s) that $\mathcal{N}_i$ can absorb at time t. Given all the bidding messages from the competing nodes, $\mathcal{N}_s$ has knowledge of the *upper bound* bandwidth assignment and will not assign any bandwidth higher than $b_i(t)$ to $\mathcal{N}_i$ at time t. Notice that it seems possible for some competing nodes to request more bandwidth than they really need; we will discuss the rational bidding values of the competing nodes in Section 4.

One important property of the RBM mechanism is that it provides *max-min fairness* [4, 5]. Suppose $\vec{x} = [x_1, \ldots, x_N]$ is the bandwidth allocation for all N competing nodes within the feasible domain $x_i \in [0, b_i]$ for $i = 1, .., N$. Then a feasible allocation is max-min fair if and only if an increase of x_i within its domain of feasible allocation must be at the cost of a decrease of some x_j, where $x_j \leq x_i$. In other words, the max-min allocation gives the competing node with the smallest bidding value the largest feasible bandwidth while not wasting any resource for the source node $\mathcal{N}_s$. From [5], one can show that there exists a unique max-min fair allocation vector $\vec{x}$, and it can be obtained by a *progressive filling algorithm*. The algorithm initializes all $x_i = 0$. It will then increase all competing nodes' bandwidth resource at the *same rate* of $1/N$, until one or several competing nodes hit their limits (i.e., $x_i = b_i$). When that happens, the resource allocation for these competing nodes will not be increased any more. The algorithm will continue to increase the resource of other competing nodes at the same rate. The algorithm terminates when all competing nodes hit their limits, or the total resource $\mathcal{W}_s$ is fully utilized. Mathematically, we can express the max-min resource distribution as follows. Let $\mathcal{N}_{\hat{1}}, .., \mathcal{N}_{\hat{N}}$ be N competing nodes sorted based on the non-decreasing value of b_i. The resource distribution of the RBM mechanism is

$$x_{\hat{k}} = \min \left\{ b_{\hat{k}}, \frac{\mathcal{W}_s - \sum_{i=1}^{\hat{k}-1} x_i}{N - \hat{k} + 1} \right\} \qquad \hat{k} = 1, \ldots, N. \quad (2)$$

Figure 4(a) illustrates the RBM with four competing nodes of $\vec{b} = [1, 2.5, 2.5, 4]$ and the resource bandwidth $\mathcal{W}_s = 7$ Mb/s. The resource allocation is $\vec{x} = [1, 2, 2, 2]$ (in Mb/s), which is depicted by the *shaded region* in the figure. Although the RBM avoids resource wastage, it does not provide any *incentive* for nodes to share information. Two competing nodes with the same value of bidding will obtain the same amount of resource regardless of their actual contribution to the P2P community.

Resource bidding mechanism with incentive (RBM-I): To provide incentive, this mechanism takes the contribution level of competing nodes into account. Let C_i be the contribution value[1] of the competing nodes $\mathcal{N}_i$, and this value reflect the amount of work that $\mathcal{N}_i$ has performed (for example, sharing and uploading files for other nodes). The contribution value C_i can be retrieved from the auditing authority $\mathcal{A}$ at the beginning of the file transfer process, or every time when the source node receives the bidding message $b_i(t)$ from the competing node $\mathcal{N}_i$.

One can enhance the resource bidding mechanism by en-

---

[1]Detail discussions about how to update and maintain the integrity of contribution values of all nodes can be found in [10].
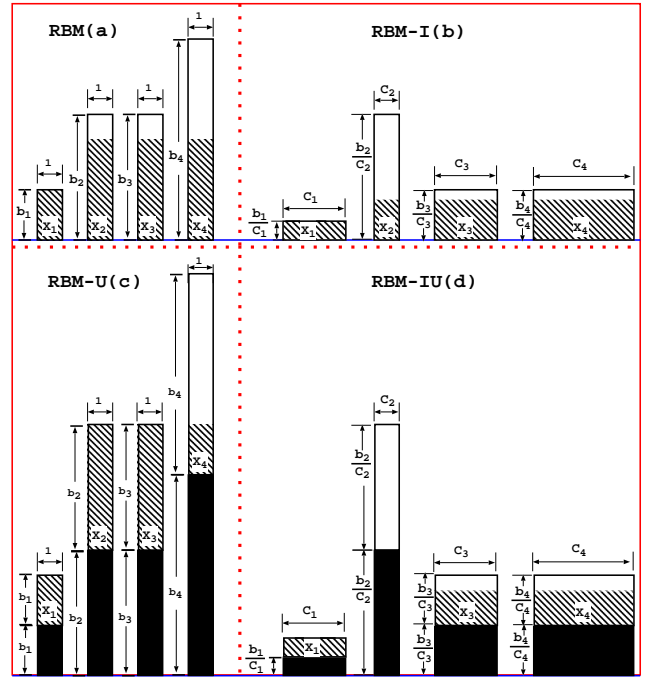

Figure 4: Resource distribution mechanisms: (a) RBM; (b) RBM-I; (c) RBM-U; (d) RBM-IU. The shaded region represents the amount of resource allocation for individual nodes.

hancing the progressive filling algorithm as follows. We distribute resources to all the competing nodes at the same time but at different rates. In particular, the competing node $\mathcal{N}_i$ will have a resource assignment rate of $C_i / \sum_{j=1}^{N} C_j$. Also, once the assigned resource to $\mathcal{N}_i$ reaches its limit of b_i, $\mathcal{N}_i$ will be taken out from the resource distribution. Therefore, one can view the mechanism as a weighted max-min resource distribution. Mathematically, we can express the RBM-I algorithm as follows. Let $\mathcal{N}_{\hat{1}}, \ldots, \mathcal{N}_{\hat{N}}$ be N competing nodes sorted based on the non-decreasing value of b_i/C_i. The resource distribution is

$$x_{\hat{k}} = \min \left\{ b_{\hat{k}}, \frac{C_{\hat{k}} \left(\mathcal{W}_s - \sum_{i=1}^{\hat{k}-1} x_i \right)}{\sum_{j=\hat{k}}^{N} C_j} \right\} \quad \hat{k} = 1, \ldots, N. (3)$$

Using the previous example in RBM but now with contributions $\vec{C} = [2.5, 1, 2.5, 4]$, the resource allocation is $\vec{x} = [1, 0.8, 2, 3.2]$ (in unit of Mb/s), as shown in Figure4(b). One important property of this mechanism is that if two competing nodes have the *same* bandwidth bidding values, then the assigned bandwidth will be *proportional* to their contribution values (i.e., $\mathcal{N}_2$ and $\mathcal{N}_3$).

Resource bidding mechanism with utility feature (RBM-U): The aim of this mechanism focuses on the efficiency of the resource allocation from the perspective of the competing nodes' satisfaction. Consider a case of two competing nodes $\mathcal{N}_i$ and $\mathcal{N}_j$ which have the same contribution values. If the bandwidth resource at the source node is $\mathcal{W}_s = 1$ Mb/s and the two bidding messages are $b_i(t) = 10$ Mb/s and $b_j(t) = 1$ Mb/s. Based on the RBM mechanism, they will each receive a bandwidth resource of 0.5 Mb/s. Although the resource at $\mathcal{N}_s$ is efficiently utilized, the degrees

of satisfaction of these two competing nodes are obviously different. To overcome this problem, we use the concept of *utility* [16] to represent the degree of satisfaction of a competing node given a certain allocated bandwidth.

We first define the family of utility functions we consider in this paper. Given an allocated bandwidth x, the utility of the node $\mathcal{N}_i$ is denoted by $U_i(x)$. The utility function we consider in this work satisfies the following three assumptions: (a) $U_i(x)$ is concave (or the marginal utility $\frac{dU_i(x)}{dx}$ is non-increasing $\forall x \geq 0$), (b) $U_i(0) = 0$, and (c) the utility depends on the ratio of $\frac{x}{b}$. In other words, $U_i(x_i) = U_j(x_j)$ whenever $\frac{x_i}{b_i} = \frac{x_j}{b_j}$ for any two competing nodes $\mathcal{N}_i$ and $\mathcal{N}_j$. The justifications for the above assumptions are as follows. First, the utility function is concave, which is often used to represent *elastic traffic* such as file transfer [16]. Concavity implies that the marginal utility is non-increasing as one increases the allocated bandwidth resource x. This captures the physical characteristics of elastic traffic: the utility increases significantly when a competing node starts receiving service. The increase of utility becomes less significant when the receiving bandwidth is nearly saturated. Second, the utility is zero when a competing node is not allocated any bandwidth. Third, because utility measures the satisfaction of a competing node, naturally, it is a function of the *ratio* of the allocated resource to the bidding resource. Furthermore, this assumption normalizes the utility of all nodes so that we can *compare* the degrees of satisfaction of different nodes.

The objective of the RBM-U mechanism is to maximize the social (or aggregate) utility. Formally, we have:

$$\max \sum_{i=1}^{N} U_i(x_i) \qquad \text{s.t.} \sum_{i=1}^{N} x_i \leq \mathcal{W}_s \ \text{ and } \ x_i \in [0, b_i] \ \forall \ i.$$

It is important to point out that the implication of this maximization problem is to allocate resource to the competing node which currently has the *largest* marginal utility (i.e., largest $dU_i(x)/dx$). The allocation process starts with $x_i = 0$ for $i = 1, \ldots, N$, then assigns resource to the node which has the largest marginal utility and ends when the resource $\mathcal{W}_s$ is used up, or all the competing nodes are fully satisfied with $x_i = b_i \forall i$.

Let us consider the following form of utility function which satisfies the above three assumptions:

$$U_i(x_i) = \log\left(\frac{x_i}{b_i} + 1\right) \qquad \text{where } x_i \in [0, b_i].$$

The marginal utility is $U_i' = (x_i + b_i)^{-1}$. Therefore, the RBM-U mechanism tries to increase the resource to the competing node which has the smallest value of $x_i + b_i$ at any time. Using the previous example of RBM of 4 competing nodes with $\vec{b} = [1, 2.5, 2.5, 4]$ and $\mathcal{W}_s = 7$ Mb/s, we use the above utility function and the resource allocation which maximizes the aggregate utility is $\vec{x} = [1, 2.5, 2.5, 1]$ (in unit of Mb/s). This result is depicted in Figure 4(c). The figure shows graphically how the mechanism works. Each competing node, say $\mathcal{N}_i$, has a lower limit height which is equal to b_i (i.e., the darkened region). The enhanced progressive filling algorithm distributes resource first to the competing node that has the *lowest* depth since that node has the *largest* marginal utility at that point. When the assigned resource to node $\mathcal{N}_i$ is equal to its maximum bidding b_i, node $\mathcal{N}_i$ is taken out from the resource distribution. The algorithm terminates when all nodes reach their maximum allocation, or when the resource $\mathcal{W}_s$ is fully utilized.

Resource bidding mechanism with incentive and utility feature (RBM-IU): One can view the RBM-IU mechanism as a *generalization* of the previous discussed mechanisms. This mechanism considers both the utilities of competing nodes and their contribution values. Each competing node, say $\mathcal{N}_i$, has its contribution value C_i and bidding message b_i. Mathematically, the RBM-IU performs the following constrained optimization:

$$\max \sum_{i=1}^{N} C_i \log\left(\frac{x_i}{b_i} + 1\right) \quad \text{s.t.} \sum_{i=1}^{N} x_i \leq \mathcal{W}_s, x_i \in [0, b_i] \ \forall \ i.$$

The RBM-IU mechanism enhances the progressive filling algorithm as follows: (a) We treat the competing node $\mathcal{N}_i$ as a bucket with area b_i and width C_i. (b) The bucket of the competing node $\mathcal{N}_i$ is located at a height of b_i/C_i; therefore the upper limit of the bucket is at a height of $2b_i/C_i$. (c) At any time, the RBM-IU mechanism increases the amount of resource put into the competing node's bucket which currently has the lowest height, i.e., the bucket that has the largest weighted marginal utility (weighted by the contribution value). It is interesting to observe that when the competing nodes have the same contribution value, the RBM-IU is equivalent to the RBM-U mechanism. The spirit of this mechanism is to increase at a rate of C_i the amount of resource given to the competing node which has the largest weighted marginal utility of $C_i/(b_i + x_i)$. Figure 4(d) illustrates the RBM-IU mechanism with $\vec{b} = [1, 2.5, 2.5, 4]$, $\vec{C} = [2.5, 1, 2.5, 4]$ and $\mathcal{W}_s = 7$ Mb/s. The final resource allocation is $\vec{x} = [1, 0, 2.3, 3.7]$ (in unit of Mb/s). From the figure, one can observe that the mechanism fills the bucket of $\mathcal{N}_i$ at most up to its area limit of b_i at the resource distribution rate of C_i. The bucket of $\mathcal{N}_i$ at the "resource level" $(x_i + b_i)/C_i$ is guaranteed to have the marginal utility $C_i/(x_i + b_i)$. The algorithm terminates when all competing nodes reach their resource limit, or when the resource $\mathcal{W}_s$ is fully utilized.

The RBM-IU mechanism can be expressed by the following pseudo-code. The source node $\mathcal{N}_s$ maintains a sorted list of competing nodes with b_i/C_i in ascending order.

RBM-IU Mechanism ()

1. **if** $(\sum_{i=1}^{N} b_i \leq \mathcal{W}_s)$ **return** $\vec{x} = \vec{b}$;/*no congestion*/
2. l=2; u=1; /*upper and lower limits index*/
3. v=C_1; w=$\mathcal{W}_s$; /* filling rate and resource capacity*/
4. level = $\frac{b_1}{C_1}$;/*initialize resource level*/
5. **while** (w> 0)
6.   **if** ($(\min\{\frac{2b_u}{C_u}, \frac{b_l}{C_l}\} - $level)*v$\geq$w)
7.   level = level + w/v; w=0;
8.   **else if** $(\frac{2b_u}{C_u} < \frac{b_l}{C_l})$
9.   w $-=$ $(\frac{2b_u}{C_u} - $level)*v; level = $\frac{2b_u}{C_u}$; v $-= C_u$; u++;
10.   **else**
11.   w $-=$ $(\frac{b_l}{C_l} - $level)*v; level = $\frac{b_l}{C_l}$; v $+= C_l$; l++;
12. **for** (each i)
13. $x_i = \min\{\max\{0, ($level $- \frac{b_i}{C_i})*C_i\}, b_i\}$;
14. **return** $\vec{x}$;

The above code performs the filling algorithm when the total bidding is greater than the total available resource. In determining the final "resource level", we have three cases in the *while* loop at line 5: (1) When the resource is used up, the loop ends with the final resource level (lines 6–7). (2) If the next available resource level is at the upper limit (or bidding level) of some competing node, we adjust the remaining amount of available resource and reduce the filling rate by that competing node's contribution value C_i since we will not give any more resource to the satisfied competing node (lines 8–9). (3) If the next available resource level is a lower limit of some competing node, then we adjust the remaining amount of available resource and increase the filling rate by that competing node's contribution value C_i (line 11). The reason is that this competing node will have the largest weighted marginal utility for its turn to gain the resource at a rate of C_i. Note that this is a *linear* algorithm with a complexity of $O(N)$, where N is the number of competing nodes at the source node $\mathcal{N}_s$. Therefore, the resource distribution can be performed efficiently.

Lastly, the following two important theorems state some of the desirable properties of the RBM-IU mechanism.

Theorem 1. *For any two competing nodes $\mathcal{N}_i$ and $\mathcal{N}_j$, the mechanism RBM-UI assigns the bandwidth resources x_i and x_j such that:*

$$if\ \frac{C_i}{b_i} \geq \frac{C_j}{b_j} \Longrightarrow U_i(x_i) \geq U_j(x_j). \tag{4}$$

Proof: Please refer to [10]. **Remarks:** The implication of this theorem is that a client which has the highest contribution per unit resource request among all the clients will receive the highest utility. Therefore, the RBM-IU provides incentive to a P2P system and increases a node's utility.

Theorem 2. *The resource allocation $\vec{x}$ is Pareto optimal, which implies that the resource allocation vector cannot be improved further without reducing the utility of at least one competing node.*

Proof: Please refer to [10].

4. RESOURCE COMPETITION GAME

In our P2P network, each competing node sends bidding messages to the source node. In return, the source node uses the mechanism RBM-IU for bandwidth resource distribution. The *interaction* between the competing nodes and the source node can be described by a game theory framework [11]. We model the interaction of resource competition as a game and explore its solution and properties. (In this paper, we focus on the competition game with one single source node only.) We also discuss how this game can be incorporated into our P2P protocol such that it converges to the Nash equilibrium.

4.1 Theoretical Competition Game

We model the resource bidding and distribution processes as a competition game between all the competing nodes. One basic postulate in game theory is that the game structure is *common knowledge* to all the players. In our game, we assume the total amount of bandwidth resource $\mathcal{W}_s$ and all the contribution values C_i's are common knowledge. This

means that all nodes know the information, know that their rivals know the information, and know that their rivals know that they know the information, and so on. Also, we only consider the non-trivial situation when $\sum b_i > \mathcal{W}_s$. The competition game can be described as follows:

1. All the competing nodes are players of the game.

2. The bidding message b_i is the strategy of the competing node $\mathcal{N}_i$. A bidding vector $\vec{b} = \{b_1, b_2, ..., b_N\}$ is a strategy profile where N is the number of competing nodes in the game.

3. The mechanism RBM-IU defines the rules and the structure of the game. We can regard mechanism RBM_IU as a mapping function which has $\vec{C}$ and $\vec{b}$ as input parameters and returns $\vec{x}$ as output.

4. The outcome of the game is the vector $\vec{x}$ which represents the amount of bandwidth resource each competing node obtains.

Lemma 1. *The mapping function RBM-IU: $\vec{C} \times \vec{b} \to \vec{x}$ is quasi-concave in each individual's strategy b_i.*

Proof: Please refer to [10].

Theorem 3. *There exists at least one Nash equilibrium in the competition game.*

Proof: Please refer to [10].

Lemma 2. *For any player, say $\mathcal{N}_i$, the strategy $b_i^* = \frac{\mathcal{W}_s C_i}{\sum_{j=1}^{N} C_j}$ implies a resource allocation of $x_i^* = \frac{\mathcal{W}_s C_i}{\sum_{j=1}^{N} C_j}$ for $i = 1, \ldots, N$.*

Proof: Please refer to [10].
Remark: The importance of the above lemma is in guaranteeing that a player can gain its fair share of resources during the competition. For some players who have small contribution values, they will not suffer from resource starvation. For free riders, however, they will eventually gain zero resource in the competition.

Theorem 4. *The strategy profile $b_i^* = \frac{\mathcal{W}_s C_i}{\sum_{j=1}^{N} C_j}$ for player $\mathcal{N}_i$, where $i = 1, \ldots, N$, is a Nash equilibrium.*

Proof: Please refer to [10].

Definition 1. *λ-collusion occurs when a subset of competing nodes $\mathcal{N}_\lambda$ use strategy profile $b_i \neq b_i^* \ \forall i \in \mathcal{N}_\lambda$, and achieve $\sum_{i \in \mathcal{N}_\lambda} x_i > \sum_{i \in \mathcal{N}_\lambda} x_i^*$.*

Theorem 5. *Assuming that all honest competing nodes use the Nash equilibrium strategy $b_i^* = \mathcal{W}_s C_i / \sum_{j=1}^{N} C_j$, the RBM-IU mechanism in the source node avoids λ-collusion.*

Proof: Please refer to [10].

4.2 Practical Competition Game Protocol

In the above sub-section, we show that the interactions between the source node and all its competing nodes can be modeled as a competition game which has a Nash equilibrium solution. This solution assigns each competing node the amount of resource *proportional* to their contributions, efficiently utilizes all resource at the source node, and prevents collusion among the group of competing nodes.

Although the theoretical competition game provides these attractive properties, there are gaps to fill in order to realize the theoretical game in an incentive P2P network. In particular, one needs to address the following problems:

- **P1** The information of contribution $\vec{C}$ and the amount of resource $\mathcal{W}_s$ is assumed to be common knowledge. How can this be implemented in a P2P system?

- **P2** In real life, a competing node, say $\mathcal{N}_i$, has its maximum download capacity, say w_i (in bits/s). Also, due to intermittent network congestion, the actual assigned bandwidth allocation x_i may be less than the actual received bandwidth x_i'. These two factors will affect the Nash equilibrium derived under the theoretical game.

- **P3** In a dynamic environment like P2P networks, new competing nodes may arrive and ask for file download, while existing competing nodes may leave due to the termination of their requested transfers. Under these situations, how can the system maintain the equilibrium point in spite of changes in the number of competing nodes?

To address these issues, let us first consider the behavior of the source node. Based on a certain strategy profile $\vec{b}$ and contribution values $\vec{C}$, the source node carries out the RBM-IU algorithm for bandwidth resource distribution. The *justification* that the source node is willing to use this mechanism is that the allocation result is Pareto-optimal (based on Theorem 2). This implies that following the RBM-IU mechanism, the source node can maximize its contribution value and enjoy better service for future file download requests. However, without perfect information for all the competing nodes, the game solution may oscillate and cause resource wastage. For the source node to maximize its contribution, it has *incentive* to help all the competing nodes reach Nash equilibrium. In our practical game protocol, the source node will signal a competing node, say $\mathcal{N}_i$, with the value of $S_i = \mathcal{W}_s C_i / \sum_{j=1}^{N} C_j$ when $\mathcal{N}_i$ initiates its request file download. This information exchange is inexpensive because: (1) the signal is sent only *once* for each competing node's arrival; (2) the signal value is computed on the fly and does not need global information about the contribution values of all the nodes in the network. Hence, issue P1 is resolved.

For the behavior of the competing nodes, let us see how the signals sent by the source node may help the game to reach its equilibrium. Suppose that a competing node, say $\mathcal{N}_i$, has the maximum download capacity of w_i and a *signal variable* s_i. Initially, s_i stores the signal value sent by the source node, i.e., $s_i = S_i = \mathcal{W}_s C_i / \sum_{j=1}^{N} C_j$. The competing node $\mathcal{N}_i$ sends its initial bidding message $b_i = \min\{w_i, s_i\}$ to the source node. After each round of data transfer, $\mathcal{N}_i$ measures x_i', the amount of bandwidth resource it receives from the source node, and stores it as the current signal value of s_i, i.e., $s_i = x_i'$, To start the next round of data transfer, $\mathcal{N}_i$ sends a new bidding message $b_i = min\{w_i, s_i\}$ to the source node. This bidding strategy assumes that the source node uses the RBM-IU mechanism, so that all competing nodes reach Nash equilibrium through feedback on their strategies. In the bidding message, competing nodes inform the source node of (1) its download bandwidth limit, and (2) whether there is any congestion along the data transfer path.

The behavior of competing nodes as described above is an attempt to resolve the issues of P2 and P3. However, one can show that using this protocol, the system may *not* be able to reach Nash equilibrium. Consider the following illustrative example. Initially the source node $\mathcal{N}_s$ has resource $\mathcal{W}_s = 6$ and it has one competing node $\mathcal{N}_1$ with $w_1 = 10$ and $C_1 = 1$. The source node sends $\mathcal{N}_1$ a signal of $S_1 = 6$. Therefore, the initial bidding message from $\mathcal{N}_1$ is $b_1 = \min\{10, 6\} = 6$ and the resource allocation is $x_1 = 6$ (which is a Nash equilibrium point). Afterwards, a new competing node $\mathcal{N}_2$ arrives with $w_2 = 1$ and $C_2 = 1$. The source node sends $\mathcal{N}_2$ a signal of $S_2 = 3$. Therefore, the initial bidding message from $\mathcal{N}_2$ is $b_2 = \min\{1, 3\} = 1$. The final resource allocation is $\vec{x} = [5, 1]$ (which is also a Nash equilibrium point). Now a new competing node $\mathcal{N}_3$ arrives with $w_3 = 10$ and $C_3 = 1$. The source node sends $\mathcal{N}_3$ a signal of $S_3 = 2$. Therefore, the initial bidding message from $\mathcal{N}_3$ is $b_3 = \min\{10, 2\} = 2$. The final resource allocation is $\vec{x} = [3, 1, 2]$. Note that this equilibrium point is *not* a Nash equilibrium since there is some degree of unfairness between the two homogeneous nodes $\mathcal{N}_1$ and $\mathcal{N}_3$, and $\mathcal{N}_3$ could have received a higher bandwidth if it had increased its bidding. Another scenario in which the final resource allocation may not be a Nash equilibrium is when some of the competing nodes experience network congestion such that $x_i' < x_i$. When these nodes, through feedback, use their new biddings $b_i = x_i'$ for the resource allocation, some of the source node resources will not be utilized and will remain idle. This condition continues even if the network congestion disappears later on for the competing nodes. In other words, the competing nodes cannot get back the amount of resource they could have obtained in the Nash equilibrium solution. To solve the problem, a competing node needs to behave more *aggressively* in order to obtain the proper amount of resource and help the system reach the new Nash equilibrium efficiently.

To properly resolve issues P2 and P3, we propose the following extension protocol. Each competing node, say $\mathcal{N}_i$, enhances its bidding by sending

$$b_i = \min\{w_i, (1+\delta)s_i\} \tag{5}$$

where δ is a small positive constant for all competing nodes. The purpose of reporting a slightly larger bidding value is to explore the possibility that there is some unused resource at the source node. The Nash equilibrium solution $\vec{x}^*$ under the theoretical model is not changed except that the strategy profile is changed to be $\vec{b}^* = (1+\delta)\vec{x}^*$. In case there are idle resources and unfair allocations temporarily in the system, competing nodes which gain a smaller amount of resource can increase their biddings and push the system to the new Nash equilibrium point. Therefore, their subsequent bidding values will increase. Eventually, a new equilibrium is reached when each competing node bids $b_i = min\{w_i, (1+\delta)s_i\}$ and receives $x_i' = s_i$.

From now on, we will assume that all competing nodes in the incentive P2P network send bidding messages according to Eq. (5). Obviously, all the competing nodes interacting with the source node will achieve a different allocation result at equilibrium than the Nash equilibrium result under the theoretical model. We classify these competing nodes into three categories at equilibrium. When the bidding is $b_i = w_i$ at equilibrium, physically the competing node receives $x_i' = w_i$, and the allocated resource must be $x_i = w_i$. It implies that the competing node does

not encounter any network congestion. When the bidding is $b_i = (1+\delta)x_i'$ at equilibrium, there are two cases to consider: (1) There is a bottleneck link (with available bandwidth v_i) between the competing node and the source node. Therefore, no matter how large the contribution value of the competing node or its bidding value, the competing node can only receive v_i amount of bandwidth resource. So we have $b_i = (1+\delta)x_i' = (1+\delta)v_i$. (2) The competing node competes with other competing nodes for the resource at the source node. Therefore, the bottleneck is on the side of the source node. So we know $b_i = (1+\delta)x_i' = (1+\delta)x_i$. Define the above three categories of competing nodes at equilibrium to be in the sets $\mathcal{N}_\alpha$, $\mathcal{N}_\beta$, and $\mathcal{N}_\gamma$, respectively.

Lemma 3. *When the dynamic game reaches equilibrium, the following equality holds:*

$$x_i/C_i = x_j/C_j$$

for all $\mathcal{N}_i, \mathcal{N}_j \in \mathcal{N}_\gamma$.

Proof: Please refer to [10].

Lemma 4. *At any equilibrium of the dynamic game, the following inequality holds:*

$$x_i/C_i + \frac{1}{2}\delta x_i/C_i \geq x_j/C_j$$

for all $\mathcal{N}_i \in \mathcal{N}_\gamma$ and $\mathcal{N}_j \in \mathcal{N}_\alpha \cup \mathcal{N}_\beta$.

Proof: Please refer to [10].

Theorem 6. *The equilibrium solution of the dynamic game described above has the bandwidth allocation solution:*

$$x_i = \begin{cases} w_i & \text{if } x_i \in \mathcal{N}_\alpha \\ v_i & \text{if } x_i \in \mathcal{N}_\beta \\ \dfrac{C_i}{\sum\limits_{j \in \mathcal{N}_\gamma} C_j}(\mathcal{W}_s - \sum\limits_{j \in \mathcal{N}_\alpha} w_j - \sum\limits_{j \in \mathcal{N}_\beta} v_j) & \text{if } x_i \in \mathcal{N}_\gamma. \end{cases}$$

In addition, it becomes a Nash equilibrium solution when δ approaches zero.

Proof: Please refer to [10].

Remark: Although the dynamic game equilibria are not strictly Nash equilibria, they are close to being Nash equilibria when δ is small. The allocation results from these equilibria are the same as the equilibrium allocation when $\delta = 0$. Therefore, we can regard the game as reaching Nash equilibrium if all the players play the Nash strategy profile.

5. EXPERIMENTS

In this section, we report experimental results to illustrate the performance and the incentive property of our resource distribution protocol. In particular, we will show that our protocol can properly adapt to dynamic join/leave of competing nodes, and to various conditions of network congestion.

Experiment A (Incentive Resource Distribution): In this experiment, we consider a source node $\mathcal{N}_s$ with resource $\mathcal{W}_s = 2$ Mb/s. There are four competing nodes $\mathcal{N}_1$ to $\mathcal{N}_4$. Their maximum download bandwidths are $\vec{w} = [2, 1.5, 1, 0.5]$ (in Mb/s). The arrival times of $\mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3$ and $\mathcal{N}_4$ are $t =$20, 40, 60 and 80 s, respectively. Unless stated otherwise, the propagation delay between a competing node and the

source node $\mathcal{N}_s$ is one second. We consider three scenarios, each using different contribution values for the four competing nodes. In **Exp. A.1**, we have $\vec{C} = [100, 100, 100, 100]$; in **Exp. A.2**, we have $\vec{C} = [400, 300, 200, 100]$; in **Exp. A.3**, we have $\vec{C} = [400, 100, 200, 300]$. Figure 5 illustrates the *instantaneous* bandwidth allocation for all the competing nodes for $t \in [0, 100]$.
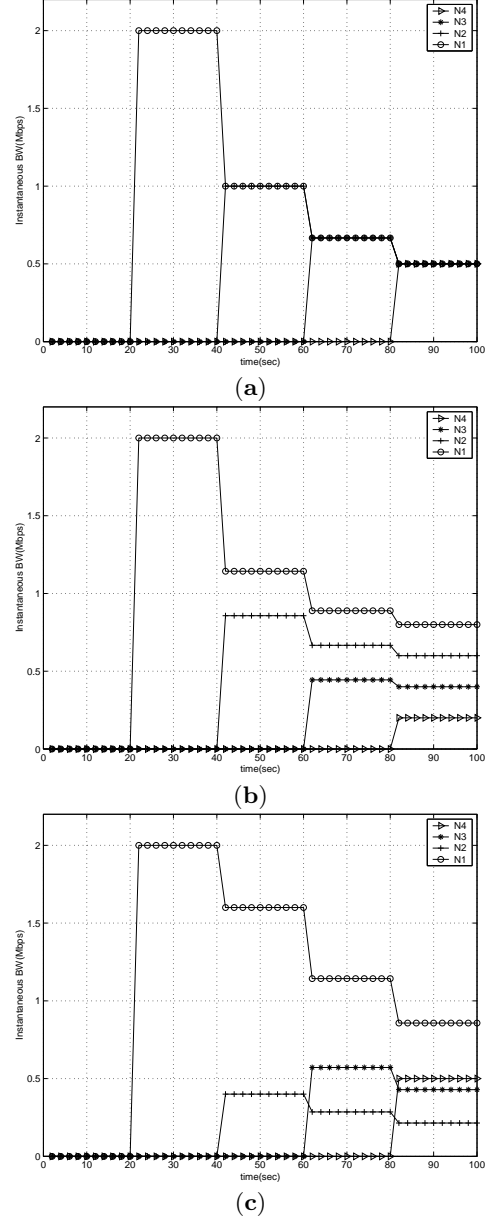
(a)

(b)

(c)

Figure 5: Instantaneous bandwidth allocations: (a) $\vec{C} = [100, 100, 100, 100]$; **(b)** $\vec{C} = [400, 300, 200, 100]$; **(c)** $\vec{C} = [400, 100, 200, 300]$.

One can make the following observations:

- Figure 5(a) shows that when all nodes have the same contribution value, they will eventually get a *fair share* (i.e., even distribution) of the bandwidth resource. For

example, for $t \in [20, 40]$, $\mathcal{N}_1$ gets all of $\mathcal{W}_s$'s resource of 2 Mb/s since it is the only competing node and $w_1 = 2$ Mb/s. For $t \in [40, 60]$, the resource is evenly shared by $\mathcal{N}_1$ and $\mathcal{N}_2$ since they have the same contribution values. When all the four competing nodes are present ($t \in [80, 100]$), each node will get a resource amount $x = 0.5$ Mb/s.

- Figure 5(b) shows that the bandwidth resource assignment is *proportional* to the contribution value of a competing node. When all four competing nodes are present ($t \in [80, 100]$), the resource allocation vector is $\vec{x} = [0.8, 0.6, 0.4, 0.2]$ (Mb/s). Hence, RBM-IU provides service differentiation, such that nodes have incentive to share information and to provide services.

- Figure 5(c) shows that the protocol will not waste any resource at the source node. Given $\vec{C} = [400, 100, 200, 300]$, the resource distribution should be $\vec{x} = [0.8, 0.2, 0.4, 0.6]$ (Mb/s). But since the maximum download bandwidth of $\mathcal{N}_4$ is $w_4 = 0.5$ Mb/s only, the remaining resource (0.1 Mb/s) will be distributed *proportionally* to $\mathcal{N}_1$, $\mathcal{N}_2$ and $\mathcal{N}_3$. The final resource distribution is $\vec{x} = [0.86, 0.21, 0.43, 0.5]$ (Mb/s).

In summary, these experiments show that the RBM-IU can provide incentive service differentiation and will efficiently utilize resources at the source node.

Experiment B (Adaptivity to dynamic join/leave of competing nodes): In this experiment, we consider one source node $\mathcal{N}_s$ with resource $\mathcal{W}_s = 2$ Mb/s. There are four competing nodes $\mathcal{N}_1$ to $\mathcal{N}_4$ with contributions $\vec{C} = [400, 300, 200, 100]$ and maximum download bandwidths $\vec{w} = [2, 1.5, 1, 0.5]$ (in Mb/s). There is a propagation delay of one second between a competing node and the source node.

We consider two scenarios of arrival and departure patterns: **Exp B.1:** $\mathcal{N}_1$ arrives and departs at $t = 40$ and $t = 160$, $\mathcal{N}_2$ arrives and departs at $t = 60$ and $t = 100$, $\mathcal{N}_3$ arrives and departs at $t = 80$ and $t = 120$, and $\mathcal{N}_4$ arrives and departs at $t = 20$ and $t = 140$. **Exp B.2:** $\mathcal{N}_1$ arrives and departs at $t = 20$ and $t = 100$, $\mathcal{N}_2$ arrives and departs at $t = 80$ and $t = 120$, $\mathcal{N}_3$ arrives and departs at $t = 60$ and $t = 140$, and $\mathcal{N}_4$ arrives and departs at $t = 40$ and $t = 160$. Figure 6 illustrates the instantaneous bandwidth allocation for time $t \in [0, 180]$.

One can make the following observations:

- The protocol can assign the proper amount of resource to competing nodes without wastage. For example, for time $t \in [20, 40]$, Figure 6(a) shows that $\mathcal{N}_4$ obtains 0.5 Mb/s (since this is its maximum download bandwidth). But for the same time period, Figure 6(b) shows that $\mathcal{N}_1$ can get 2.0 Mb/s, its maximum download bandwidth and the full resource of the source node.

- Both Figures 6(a) and (b) show that the protocol can fully utilize the source resources. For example, for period $t \in [40, 120]$, the source node distributes the resource proportionally to the contribution values of the competing nodes. The assignment is independent of the number of competing nodes and their arrival patterns.

- The protocol can reach the same equilibrium point, *independent* of the arrival and departure sequences of

Exp B.1 or **Exp B.2**. For example, consider the time period $t \in [80, 100]$. The resource distribution for both cases is $\vec{x} = [0.8, 0.6, 0.4, 0.2]$ (in Mb/s), which is also the Nash equilibrium point.

In summary, these experiments show that the protocol is adaptive to the arrival and departure sequence, and it provides service differentiation to different competing nodes having different contribution values.
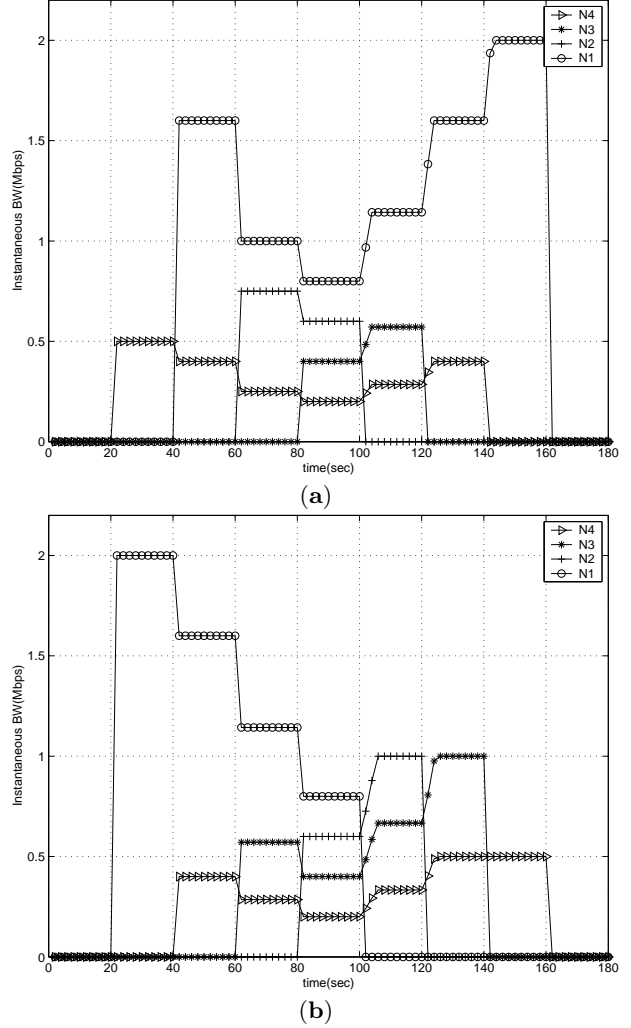
(a)

(b)

Figure 6: Instantaneous bandwidth allocations for arrival and departure patterns (a) Exp B.1; (b) Exp B.2.

Experiment C (Adaptivity to network congestion): In this experiment, we consider one source node $\mathcal{N}_s$ with resource $\mathcal{W}_s = 2$ Mb/s. At time $t = 0$, there are already four competing nodes $\mathcal{N}_1$ to $\mathcal{N}_4$ in the system. These nodes have contribution values $\vec{C} = [400, 300, 200, 100]$ and maximum download bandwidths of $\vec{w} = [2, 1.5, 1, 0.5]$ (in Mb/s). There is a propagation delay of one second from each competing node to the source node. In this experiment, we consider the dynamic congestion situation. In particular, the congestion occurs along the communication path between $\mathcal{N}_1$ and the source node $\mathcal{N}_s$. Congestion occurs twice, at time

$t = [30, 40]$ and at time $t = [50, 60]$. During the congestion, the available bandwidth along the communication path is reduced to 400 kb/s.
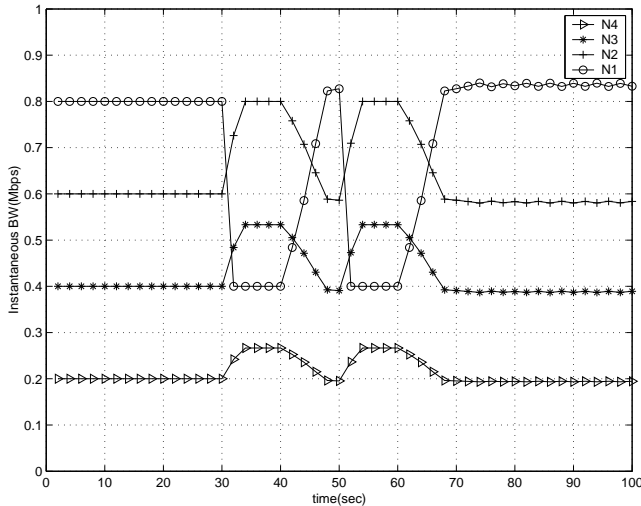

Figure 7: Instantaneous bandwidth allocations for four competing nodes; congestion occurs at $t = [30, 40]$ **and** $t = [50, 60]$**.**

Figure 7 illustrates the instantaneous bandwidth allocation of all four competing nodes for time $t \in [0, 100]$. One can make the following observations:

- At time $t = 0$, the system starts at Nash equilibrium with resource allocation of $\vec{x} = [0.8, 0.6, 0.4, 0.2]$ (in Mb/s).

- Between time $t \in [30, 40]$ (or $t = [50, 60]$), since there is network congestion, the competing node $\mathcal{N}_1$ receives less transfer bandwidth from the source node. Other competing nodes $\mathcal{N}_2$ to $\mathcal{N}_4$ can discover this idle bandwidth resource of 0.4 Mb/s via their bidding messages. The source node $\mathcal{N}_s$ will distribute this excessive bandwidth resource to the other three competing nodes proportionally to their contribution values. New Nash equilibria are reached ($t \in [35 - 40]$ and $t \in [55 - 60]$).

- When the congestion disappears, the competing node $\mathcal{N}_1$ can gain back its proper resource amount of $x_1 = 0.8$ Mb/s. Also, the new Nash equilibrium can be quickly reached and the final resource allocation is $\vec{x} = [0.8, 0.6, 0.4, 0.2]$ Mb/s.

In summary, this experiment shows that the protocol is adaptive to network congestion. During network congestion, the resource at the source node will not be wasted but rather distributed proportionally to other competing nodes.

6. CONCLUSION

In this paper, we present a P2P network protocol that provides service differentiation and gives incentive for nodes to share information and provide services. The solution consists of the resource allocation mechanism RBM-IU and the interaction protocol for competing nodes to reach equilibria of the competition game induced by RBM-IU. The solution is efficient: (1) RBM-IU can be implemented by a linear time algorithm, (2) the feedback based bidding messages used by competing nodes are simple, and (3) RBM-IU achieves Pareto-optimal allocation results. The robustness of the solution is evidenced by the fact that all competing nodes can reach the equilibrium solutions of the competition game. The justification for the source node to use our protocol is its guarantee of Pareto optimality. On the other hand, competing nodes are motivated to use the protocol because it guarantees Nash equilibrium. We also show that the protocol is adaptive to various node arrival and departure events, and to different forms of network congestion.

7. REFERENCES

[1] The Gnutella Protocol Specification v0.4 1, document revision 1.2.

[2] Limewire : a Gnutella client software.

[3] E. Adar and B. Huberman. Free riding on Gnutella. *FirstMonday*, 2000.

[4] D. Bertsekas and R. Gallager. *Data Networks*. Prentice-Hall, Englewood Cliffs, New Jersey, 1992.

[5] J.-Y. Boudec. Rate adaptation, congestion control and fairness: A tutorial, http://icapeople.epfl.ch/leboudec.

[6] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong. Freenet: A distributed anonymous information storage and retrieval system. *Lecture Notes in Computer Science*, 2001.

[7] A. C. Fuqua, T. Ngan, and D. S. Wallach. Economic behavior of peer-to-peer storage networks. *Workshop on Economics of Peer-to-Peer Systems, June 2003*.

[8] P. Golle, K. Leyton-Brown, I. Mironov, and M. Lillibridge. Incentives for sharing in peer-to-peer networks. *Proceedings of the ACM Conference on Electronic Commerce*, 2001.

[9] G. Hardin. The tragedy of the commons. *Science*, 162:1243–1248, 1968.

[10] R. Ma, S. Lee, J. Lui, and D. Yau. Incentive P2P Networks: Theory and implementation. *Technical report of Dept. of CSE, Chinese University of Hong Kong*.

[11] A. Mas-Colell, M. D. Whinston, and J. R. Green. *Microeconomic theory*. Oxford University Press, 1995.

[12] N. Nisan and A. Ronen. Algorithmic mechanism design, 1999.

[13] D. Parkes. Chapter 2, iterative combinatorial auctions: Achieving economic and computational efficiency ph.d. dissertation, univesity of pennsylvania, May, 2001.

[14] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content addressable network. In *Proceedings of ACM SIGCOMM*, 2001.

[15] A. Rowstron and P. Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. *Lecture Notes in Computer Science*, 2218, 2001.

[16] S. Shenker. Fundamental design issues for the future internet. *IEEE Journal on Selected Areas in Communication*, 13(7), September 1995.

[17] J. Shneidman and D. Parkes. Rationality and self-interest in peer to peer networks. *International Workshop on Peer-to-Peer Systems (IPTPS)*, 2003.

[18] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. *ACM SIGCOMM*, 2001.

[19] W. Wang and B. Li. To play or to control: a game-based control-theoretic approach to peer-to-peer incentive engineering. *IWQoS, 2003*.

[20] B. Y. Zhao, J. D. Kubiatowicz, and A. D. Joseph. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Technical Report UCB/CSD-01-1141, UC Berkeley, Apr. 2001.

[21] S. Zhong, Y. Yang, and J. Chen. Sprite: A simple, cheat-proof, credit-based system for mobile ad hoc networks, 2002.