A Mechanism for Pricing and Resource Allocation in Peer-to-Peer Networks

This version: July 2010

Chetan Kumar\textsuperscript{a} ● Kemal Altinkemer\textsuperscript{b} ● Prabuddha De\textsuperscript{b}

\textsuperscript{a}College of Business Administration, California State University San Marcos

\textsuperscript{b}Krannert School of Management, Purdue University

ckumar@csusm.edu (Corresponding author. Phone: 760 477 3976. Website: www.csusm.edu/ckumar), kemal@purdue.edu, pde@purdue.edu

Abstract

In this study, we design a pricing and allocation mechanism for a peer-to-peer (P2P) network that allows users in a firm to effectively share their computing resources. This mechanism allows tasks (incoming jobs) to be allocated to resources (participating peers) in an organization in a decentralized manner. The base case of our P2P mechanism derives the optimal transfer price that maximizes the net value, which is characterized as the difference between the expected gross value of the jobs and the expected delay cost, for the peers in the network. The optimal price for executing any job at a peer location is essentially equal to the marginal delay cost it imposes on all current jobs at that node. With this pricing scheme, no individual user has an incentive to overutilize shared resources, thereby avoiding the ‘Tragedy of the Commons’ for the P2P network. Our model builds on the classical Mendelson (1985) study that was one of the first to look at the control and management issues related to a single server computer system. In this study, we model transfer pricing for a multiple server environment of a P2P network. The original Mendelson (1985) model thus becomes a special single server case of our general model. Our basic model is extended to incorporate situations where the peers have queue length constraints, which may be used for providing Quality of Service (QoS) guarantees to users. We then perform numerical computations that illustrate the effects of job arrivals on prices and job allocations at the individual servers. There is an enormous potential for P2P based technologies to help organizations manage their computer resources effectively. Therefore, we believe that this study contributes to an important area for information systems (IS) research.

Keywords: Peer-to-Peer Networks, Pricing Mechanism, Resource Allocation, Resource Sharing, Managing Computer Networks

\textsuperscript{1} The authors thank the seminar participants of the Workshop on Information Systems and Economics (WISE) 2005, Purdue University, Rob Kaufman, and anonymous referees, for their valuable remarks on this study.
1 Introduction and Problem Motivation

Peer-to-Peer (P2P) network based technologies allow a distributed community of users to share a variety of digital or computer processing resources. The novelty of P2P networks with respect to traditional client-server networks is that they do not necessarily require centralized or dedicated servers (Krishnan et al. 2003). Every node, or peer, that forms a part of the network may potentially contribute resources to other peers. As a result, P2P networks have many advantages over centralized networks, such as: (a) inherent scalability, (b) no single point of failure, and (c) self-administration capabilities (Li et al. 2002).

Figure 1 shows an example of a P2P network that can be used for sharing computer resources. Any peer can communicate with other nodes in the network. A task that is required by one peer can be potentially allocated to other nodes, thereby allowing the peers to pool their resources.

P2P system users combine and utilize their resources. Therefore, the P2P network may be considered as a ‘public good’ or shared utility (Krishnan et al. 2003). Hardin (1968) pointed out in his influential study that some public goods can be overutilized due to selfish behavior of individuals. For example, a common livestock grazing facility for farmers is often overgrazed in the absence of any regulation. This phenomenon is called the ‘Tragedy of the Commons’ and the concept may be applied to different shared utility environments such as the earth’s natural resources (Smith 1981, Rothbard 1982).

An interesting aspect of a P2P environment, versus traditional public good examples, is that any peer can impact other peers in the network both positively and negatively. The effect that individuals have on others, also called externality, has been empirically observed specifically in P2P file sharing networks (Asvanund et al. 2004). More peers can increase congestion that slows down the network. On the other hand, each peer may also contribute their own resources that benefit other users. Consequently, we can substantially mitigate the ‘Tragedy of the Commons’ for P2P networks by providing incentives for individual users to utilize the system responsibly. The objective of this study is to design such a mechanism for a general P2P network environment where users in a firm may share any computing resources.
P2P technology-based applications have become increasingly popular with businesses and individuals in recent times. This trend has accelerated in the Internet era (Ottolenghi 2007). Besides the well known and (il)legal P2P file sharing networks such as Kazaa, the first version of Napster, and Gnutella, P2P technologies are used in a wide variety of business settings (McDougall 2000, Jones 2006). Some common P2P applications are as follows: content delivery networks (CDN) such as Akamai (www.akamai.com), collaboration tools such as Groove (www.groove.net), cooperative cache networks such as IRCache (www.ircache.net), consumer-to-consumer ecommerce mechanisms such as Ebay (www.ebay.com), distributed processing services such as Tivoli (www.ibm.com/tivoli), knowledge management, etc. (http://en.wikipedia.org/wiki/peer-to-peer). Some of the largest IT companies such as Microsoft, HP, and IBM have invested significant resources in such P2P applications (Ottolenghi 2007).

For many organizations, their ability to effectively deliver services to end users is critically dependent on how well they manage their computer networks. Examples include firms such as Akamai, or Reuters financial network, that deliver digital content over the Internet and other computer networks. Even for organizations that may not directly deal with digital products, such networks are crucial for managing their supply chains and for disseminating information. However, with the Internet getting ever more popular we are already experiencing congestion that leads to increased waiting time for users (Datta et al. 2003). P2P based technologies have enormous potential for helping organizations manage their computer resources effectively. Therefore, it is an important area for information systems (IS) research.

There is considerable research focused primarily on the performance analysis of computer networks and systems (Kleinrock 1964, Tanenbaum 2003). Mendelson’s (1985) classical work was one of the first to look at the control and management issues related to a single-server computer system. It provides a microeconomic analysis that incorporates queuing effects and incentive issues. One of the key results of the study, which looks at transfer pricing for a single-server system, is that the marginal job should be priced at the level of congestion it imposes on all the jobs in the system. Mendelson and Whang (1990) extend the model for the case of jobs with multiple priorities and heterogeneous sizes. The
main concern of our paper is to study the pricing issue for congestible resources from a broader perspective of a P2P system. This involves examining a multiple-server computer system where the servers within an organization, which may be located at different coordinates, behave as peers that can share their computing resources. The original Mendelson (1985) model thus becomes a special single-server case of our general model.

This study focuses on the resource allocation applications of P2P technologies. The objective is to design a pricing and allocation mechanism for a P2P network that allows users in a firm to effectively share their computing resources. This mechanism, which assumes a specific network structure, allows tasks (incoming jobs) to be allocated to resources (participating peers) in a decentralized manner. The tasks are allocated so that no individual user has an incentive to overutilize shared resources, thereby avoiding the ‘Tragedy of the commons’ for the P2P network. The performance of this decentralized approach (in terms of overall delay and utilization levels observed in the system) can be compared to that of a centralized mechanism.

The mechanism may also be characterized under different peer objective functions. Instances of different peer objective functions include the following: maximizing the net value of computer services to the organization as a whole (net value maximization), maximizing the profit of the computing service provider (profit maximization), and minimizing the delay cost of end users (delay cost minimization). We consider the case where peers maximize the net value of sharing resources in the network. This objective is appropriate for an organizational setting where the firm is interested in maximizing benefit to the overall network. The network consists of multiple servers or peers, with many jobs arriving at these peers. Here the firm owns the servers, and its objective is to maximize the net value of all network users. This is in contrast to P2P file sharing networks composed of individual users, outside specific organizations, that typically do not consider any overall network benefit objectives. We also extend our mechanism to consider a case where every peer in the network has a maximum permissible queue length constraint. This would correspond to having a quality of service (QoS) guarantee since user delays would
now be limited to a specific amount. Using this we can consider the case where any server has the ability to outsource a job to a third party when the utilization level at that server reaches a certain point. Our mechanism provides a beneficial method for organizations to better utilize and share their resources, given the inherent advantages of decentralized P2P networks.

The plan of the rest of this paper is as follows. We first discuss the literature related to our topic. Next we present the analytical model. The model section consists of two subparts: We first present the base case of our model, followed by an extension to the basic model. We then illustrate the model with numerical computations. Finally, we discuss the conclusions and areas for future research.

2 Related Literature

There has been a substantial literature on computer networks since their introduction in the 1960s. Kleinrock (1964) first reported on communication networks by linking multiple computers that could transfer data. This eventually led to the architecture of the present day Internet (Kleinrock 2002). Most technical studies on computer networks focus on performance analysis issues (Kleinrock 1976, Tanenbaum 2003). These include network architecture and topology design for achieving system performance goals. Readers are referred to Tanenbaum (2003) for an extensive discussion on computer networks. Mendelson (1985) examines the control and management issues related to a computer system in his classical study. He develops a transfer pricing model for equal-sized jobs arriving at a single-server environment. Mendelson and Whang (1990) then extend the model to include multiple priorities and heterogeneous job sizes. In this paper, we model a multiple server environment of a P2P network where the users may share their computing resources. The multiple server setup of our model introduces an additional level of complexity to the basic Mendelson (1985) model, similar to the one introduced by the multiple priority model of Mendelson and Whang (1990). However, both the earlier models (Mendelson 1985, Mendelson and Whang 1990) ignore higher order interaction effects due to locations on pricing
because their settings involve only a single server. The original Mendelson (1985) model thus becomes a special single server case of our general model.

In addition to drawing on the work in IS mentioned above, we also draw on results from various related research in operations research and economics. Specifically, we apply the well-known Little’s Law (which states that the average length of a queue in a system at the steady state is equal to the mean arrival rate times the mean waiting time per job) for modeling the steady state of our system (Kleinrock 1976, Kleinrock 2002). MacKie-Mason and Varian (1995) explore the implications of congestion pricing for the Internet under centrally-planned, competitive, and monopolistic environments. Along those lines, we consider the case where peers have the objective of maximizing the net value of sharing resources in the network. Gupta et al. (1997) have also considered net value maximization objective for Internet traffic pricing. In addition, we extend the model to include constraints that allow users to have a guaranteed level of QoS. This is analogous to Hsu et al. (2008) that have examined incentive compatible pricing with QoS guarantees for a service system.

Other studies have also considered variations of the individual characteristics of our model. However, their results are specific to the computer environment being considered. Ernst et al. (2006) model a task allocation problem for minimizing cost of assigning tasks to a set of processors. The study focuses on a centralized decision making environment and does not consider a pricing mechanism. Grid computing is a related stream of research for sharing computing power for resource-intensive tasks. Bapna et al. (2008) develop a market-based resource allocation model using a combinatorial call auction approach. Parpas and Rustem (2008) propose an auction mechanism for decentralized resource allocation in grid computing. Cody et al. (2008) survey security issues in grid computing and discuss its importance for wide-spread adoption by businesses. Kumar et al. (2009) develop heuristics for real-time scheduling of grid computing. Zhang et al. (2008) study the dynamics of grid computing under a monopolistic service provider scenario using real option valuation. However, grid computing studies focus specifically on sharing extremely high computing power for relatively short periods of time.
Ercetin and Tassiulas (2003) construct a market based mechanism for minimizing user delays in a network where CDN proxy caches and content providers are behaving selfishly. Hosanagar et al. (2005) further develop a pricing scheme for caching, involving content providers and cache operators, with multiple levels of QoS. However, both the latter studies are specific to a caching environment where copies of web objects are placed at locations that are closer to end users in order to reduce access delays. In this study, we model a pricing and allocation mechanism for a decentralized, multi server environment where peers may share any computing resources. Therefore, our mechanism is applicable for a number of business settings where P2P based technologies may be utilized (Ottolenghi 2007). For many organizations, their ability to effectively deliver services to end users is critically dependent on how well they manage their computer networks. We believe that our P2P network mechanism is useful for satisfying this organizational emphasis.

Though the concept of P2P based applications have gained popularity, IS literature in that area is scarce due to its relative novelty. Li et al. (2002) demonstrate the advantages of scalability of P2P networks compared to centralized networks. Krishnan et al. (2003) identify the special characteristic of P2P file sharing networks versus other public goods or shared utilities. Asvanund et al. (2004) empirically demonstrate the presence of both positive and negative network externalities in P2P music sharing networks. Bhattacharjee et al. (2007) study the effect of digital sharing technologies on survival of music albums on ranking charts. The focus of the earlier studies was the popular P2P file sharing networks. Buragohain et al. (2003) and Feldman et al. (2004), among others, have proposed methods to reduce lack of user contribution in such networks (Feldman and Chuang 2005). The issue of P2P networks composed of individual users, outside specific organizations, that are behaving strategically is discussed in more detail later. Many file sharing networks have subsequently run into legal issues (Jones 2006). However, P2P based applications have a wide variety of applications and are being increasingly adopted by businesses (Ottolenghi 2007). In this study, we go beyond by considering a general P2P environment where users in a firm can share any computing resources.
The topic of public goods has been widely studied in economics, science, and management areas. Samuelson (1954) first identified the concept of the public good environment where users are non-excludable. Hardin (1968) characterized as a ‘Tragedy of the Commons’ the outcome where some public goods may be overutilized due to selfish behavior of individuals. This influential concept has been applied to many different settings such as air pollution (Rothbard 1982), earth’s natural resources (Smith 1981), evolutionary biology (Francisco and Gordo 2006), and Internet infrastructure (Gupta et al. 1997). While P2P networks can also be considered as public goods, they have an interesting characteristic in that every peer has both a positive and a negative effect on other users (Krishnan et al. 2003). Consequently, we can substantially mitigate the ‘Tragedy of the Commons’ for P2P networks by providing incentives for individual users to utilize the system responsibly. The objective of this study is to design such a mechanism.

3 Model

Our P2P resource-sharing mechanism models the steady-state behavior of a multiple server system. This abstraction allows us to arrive at insights that are broadly applicable to an actual P2P system. We have a number of assumptions about our model parameters for simplicity. In addition, we are able to capture the relevant characteristics of a decentralized P2P system. This is consistent with existing studies on management of computer networks, including Mendelson (1985) and Mendelson and Whang (1990). The assumptions for the model are: (a) there are multiple servers in the network in an organization, where each node is modeled as an M/M/1 queueing system that faces an exogenous Poisson job arrival process, (b) the jobs arriving at any server may be allocated to any of its ‘direct neighbor’ servers that have direct network connections with the server under consideration (the direct neighbors for any given network may have evolved out of servers agreeing to share their computing resources), (c) jobs move instantaneously between neighbor servers, (d) all incoming jobs at the various servers are of the same size, and (e) every job makes an individual, decentralized decision about joining or not joining the system (the decision of
the marginal job does not affect the overall arrival rate at any server). Using these characteristics we can arrive at analytical conclusions that may then be applied to any multi server P2P environment.

We first discuss the base case of our model, which develops transfer pricing for a multiple server P2P system in a firm, in the following sub-sections. We then extend the basic model to include a queue length constraint that provides quality of service (QoS) guarantees for the system. Next we briefly discuss studies on P2P file sharing networks composed of individual users, outside specific organizations, that are behaving strategically.

3.1 Base Case

In the model base case, our objective is to derive an expression for the optimal incentive-compatible transfer price that any arriving job at a node pays at the steady state. There are numerous business examples where this pricing scheme would be applicable. One such example is IBM providing IT infrastructure management services to clients (www.ibm.com). Another example is a CDN provider such as Akamai (www.akamai.com). These companies have servers or peers at different locations that have a variety of computing tasks being requested for execution, as shown in Figure 1 earlier. The requests may arrive from internal sources or external clients. How should server locations price and allocate the jobs such that, given the objective of the peers, the outcome is optimal for the network? The same question is applicable when independent servers within a firm are linked to form a P2P network. The final pricing and job allocation decisions in the network impacts the service experienced by customers in terms of waiting time.

We consider a P2P network of queues where every location observes a fixed exogenous arrival rate of jobs. We develop a pricing scheme where an individual user makes a decision whether to join the system on the basis of a posted price. A job may be executed either at the location where it has arrived, or it may be allocated to a neighbor location that is connected by a direct link. We characterize the model as follows: Let $j$ represent any server location and $i$ be a neighbor of $j$. The effective arrival rate of jobs at any server $j$ is $\Gamma_j$. This may also be considered the total allocation of jobs at any location, given the
server loads in the system. If $\lambda_i$ is the arrival rate of a job originating at $i$ to be executed at $j$, then

$$\Gamma_j = \sum_i \lambda_i.$$ 

We assume that this effective arrival or total allocation of jobs $\Gamma_j$, which is the sum total of fractional arrival rates from different locations, follows a Poisson process. Note that $\Gamma_j$ includes jobs that have originally arrived at $j$ and are executed at $j$ itself. The congestion level at different locations eventually determines where a job is executed irrespective of where the job originates.

Figure 2 provides a graphical representation of a simple P2P network with three server locations $i$, $j$, and $k$ that are all neighbors of one another. The solid arrows represent $\lambda_i$, $\lambda_j$, and $\lambda_k$, the exogenous job arrivals at each location. The dotted arrows represent $\lambda_j$, $\lambda_j$, and $\lambda_j$, the jobs allocated for execution at location $j$ and $\Gamma_j = \sum_i \lambda_i$. The objective of the model is to obtain the optimal $\Gamma^* = \sum_i \lambda_i^*$. 

Let $\Lambda\left(\Gamma_j\right)$ represent a monotonically increasing, continuously differentiable, and strictly concave value function of the gross value gained by system users per unit of time at $j$ when the arrival rate of jobs is $\Gamma_j$. This is a standard assumption in literature for valuing congestible resources (Mendelson 1985, Mendelson and Whang 1990, MacKie-Mason and Varian 1995). As the number of users increase there is a diminishing return to benefit accrued to system users due increased congestion and waiting times. The gross value of jobs can be considered the utility gained by users for executing jobs. This eventually translates to the amount that users are to pay for services. We assume that there are no fixed costs associated with setting up the decentralized P2P network. Including a onetime set up cost does not affect the optimal solution. Hence, it is not included in our model.

Let $\nu$ be the per-unit delay cost for any job. This is consistent for all users at all locations. We assume that the delay cost is linear in time. The linear cost structure is useful for modeling simplicity, and is also used in Mendelson (1985) and Mendelson and Whang (1990). It can also serve as a base for comparison in situations where unit delay costs are non-linear. A user is charged a price $p_j$ for any job
that is executed at node $j$. Therefore, the total cost incurred by the user consists of two components: the price $p_j$ paid for the job, and an additional cost due to the job’s delay. Let $W_j(\Gamma_j)$ denote the expected time a job spends at $j$ when the total effective arrival rate is $\Gamma_j$. The expected delay cost faced by an incoming job at $j$ is $vW_j(\Gamma_j)$. Therefore, the user’s total expected cost is $p_j + vW_j(\Gamma_j)$. A user will choose to submit a job if and only if its value exceeds the total expected cost. Hence, at the steady state, the marginal user whose job value $\Lambda'(\Gamma_j)$ is equal to the total expected cost will be indifferent between joining and not joining the system. This relationship is represented as follows:

$$\Lambda'(\Gamma_j) = p_j + vW_j(\Gamma_j).$$

(1)

We assume that the system satisfies Little’s Law, that is, given an effective arrival rate or job allocations of $\Gamma_j$ at $j$, the expected job queue length $L_j(\Gamma_j)$ is $\Gamma_jW_j(\Gamma_j)$. This also implies that $W_j(\Gamma_j)$ is a strictly increasing and differentiable function of $\Gamma_j$. Additionally, as the utilization of the system tends to one, the expected waiting time of jobs in the system tends to infinity. System utilization is given by the ratio $\Gamma_j/\mu_j$, where $\mu_j$ is the processing capacity of any node $j$. We discuss in detail later that waiting time for a job at $j$ is $W_j=1/(\mu_j-\Gamma_j)$. Since the waiting cost per job per unit time is $v$, the expected delay cost incurred by the system users per unit time is $vL_j(\Gamma_j)$. The marginal delay cost due to a job arrival can be obtained by differentiating the expected delay cost with respect to $\Gamma_j$, which yields:

$$v\left(\frac{\partial L_j}{\partial \Gamma_j}\right) = vW_j(\Gamma_j) + v\Gamma_j\left(\frac{\partial W_j}{\partial \Gamma_j}\right).$$

(2)

\[\downarrow\text{Marginal system delay cost} \quad \downarrow\text{Delay cost for user due to own work load} \quad \downarrow\text{Added workload delay cost for every other system user}\]
In equation (2), the marginal system delay cost expression can be considered as being comprised of a self-regulating term $vW_j(\Gamma_j)$ and a negative externality term $v\Gamma_j(\partial W_j/\partial \Gamma_j)$. The first term is self-regulating as it is the delay cost that a user experiences due to her job’s work load. The second term is a negative externality as it is the added work load delay cost that every other system user experiences due to a submitted job.

In order to determine the optimal transfer prices, every peer maximizes its expected ‘net value,’ which is characterized as the difference between the expected gross value of the jobs and the expected delay cost. Note that every peer $j$ may execute jobs of users that have originated at $j$ or have been allocated from other locations $i \neq j$. Given a fixed capacity, the price $p_j$ uniquely determines the arrival rate $\Gamma_j$, and the net-value maximization problem may be written as:

$$\max_{\Gamma_j} \left\{ \Lambda(\Gamma_j) - vL_j(\Gamma_j) \right\} \quad \text{The Net-Value Maximization Problem.} \quad (3)$$

Note that the capacity constraint $\Gamma_j < \mu_j$ does not appear in equation (3) as this condition is implicitly assumed to be inviolable. This is because as $\Gamma_j$ approaches $\mu_j$, the delay cost rapidly approaches infinity. A more detailed explanation for the above condition is provided in the following sections.

**Proposition 1 (The P2P Job Execution Optimal Price Proposition).** Under the net value maximization objective of peers, the optimal price for executing a job at location $j$ is given by:

$$p_j^* = v\Gamma_j^*(\partial W_j/\partial \Gamma_j^*). \quad (4)$$

where $\partial W_j/\partial \Gamma_j^*$, the incremental delay added to $W_j(\Gamma_j)$ as a result of an incremental arrival $\Gamma_j^*$, is evaluated at the optimal arrival rate of $\Gamma_j^*$.

**Proof.** The first-order condition (FOC) for maximizing equation (3) is:

$$\Lambda'(\Gamma_j^*) = v\left( \partial L_j/\partial \Gamma_j^* \right). \quad (5)$$
Equation (5) provides the optimal arrival rate $\Gamma_j^*$ that maximizes the system net value. Note that $\Gamma_j^*$ may consist of jobs exogenously arriving at $j$ as well as allocations from other locations $i \neq j$. In order to determine the net value maximizing price $p_j^*$, we substitute equations (1) and (2) into (5) to obtain (4).

The optimal price $p_j^*$ is equal to the externality cost, which is the expected delay cost per unit time inflicted on the rest of the system by an infinitesimal increase in job flow. Therefore, the optimal price for executing any job is essentially equal to the marginal delay cost it imposes on all current jobs in the system. Note that every node $j$ has a specific price $p_j^*$ for executing the jobs that arrive at it. This is a generalization of the original Mendelson (1985) model which does not consider multiple server locations. With this pricing scheme, no individual user has an incentive to overutilize the system. Therefore, it can be used to mitigate the ‘Tragedy of the Commons’ for P2P networks. For consistency, henceforth all optimal expressions for price $p_j^*$, and arrival rates $\sum_i \lambda_i^*$ or job allocations $\Gamma_j^*$, are represented with an asterisk $^*$.

The optimal price may be expressed in terms of the capacity $\mu_j$ and job arrival rate $\Gamma_j$ at any server $j$. This allows us to calculate numerical values for resource allocation decisions that achieve the objective of peers in the network. An application of this is demonstrated in the numerical computations section. By definition, the waiting time $W_j$ for a job at server $j$ is:

$$W_j = 1/\left(\mu_j - \Gamma_j\right).\quad (6)$$

Note that $W_j$ is an exponential function and that it tends to infinity as the job arrival rate $\Gamma_j$ approaches the capacity $\mu_j$. Thus, $\mu_j > \Gamma_j$ in order to avoid an infinite delay for jobs in the P2P network. Differentiating (6) with respect to $\Gamma_j$ gives us:

$$\partial W_j / \partial \Gamma_j = 1/\left(\mu_j - \Gamma_j\right)^2.\quad (7)$$
Substituting equation (7) in (4) and replacing $\Gamma_j^*$ with $\sum_i \lambda_i^*$, we can rewrite the expression for the optimal price as:

$$p_j^* = v \left( \sum_i \lambda_i^* \right) \left[ \frac{1}{\left( \mu_j - \sum_i \lambda_i^* \right)^2} \right].$$

(8)

From equation (8), we infer that as the effective arrival rate $\sum_i \lambda_i^*$ at $j$ approaches the capacity $\mu_j$, the denominator in the expression approaches zero and the price $p_j^*$ approaches infinity. The non-linear nature of $\partial W_j / \partial \Gamma_j$ in equation (4) is captured by $1/\left( \mu_j - \Gamma_j \right)^2$ in equation (8). The aggregate effect on price due to the effective arrival rate $\sum_i \lambda_i^*$ at $j$ may be contrasted against that of the marginal effect on price due to a marginal arrival $\lambda_i^*$ from any $i \neq j$. Our P2P network of queues consists of job arrivals from multiple sources. Therefore, we need to consider the interaction effects of job arrivals from different sources on the price for executing a job at any direct neighbor location. This is obtained by considering the higher order comparative statics analysis of the optimal price function (8), as shown next.

We first consider the case where a marginal job arrives at $j$ from a direct neighbor location $i \neq j$. The first order effect on the optimal price due to the marginal arrival $\lambda_i$ is obtained by differentiating equation (8) with respect to $\lambda_i$ as follows:

$$\partial p_j^* / \partial \lambda_i = v \left[ \frac{1}{\left( \mu_j - \sum_i \lambda_i^* \right)^2} + 2 \left( \sum_i \lambda_i^* \right) \left/ \left( \mu_j - \sum_i \lambda_i^* \right)^3 \right]\right].$$

(9)

With an increase in $\lambda_i$, the denominator terms of equation (9) decrease and the numerator term increases. Overall, there is a positive effect on marginal price due to a marginal increase in $\lambda_i$. Note that every marginal job arrival has only a first order effect on prices; therefore, $\partial^2 p_j^* / \partial \lambda_i^2$ and higher order effects are assumed to be negligible.
Next, we consider the second order interaction effect on change in marginal price due to a marginal arrival from any other direct neighbor location \( k \neq i \neq j \). This is obtained by differentiating (9) further with respect to \( \lambda_{sj} \) as follows:

\[
\frac{\partial^2 p_j^*}{\partial \lambda_y \partial \lambda_{sj}} = \nu \left[ \frac{4n}{\mu_j - \sum_i \lambda_{ij}^*} + \frac{2n}{\mu_j - \sum_i \lambda_{ij}^*} \right].
\]  

(10)

As in the first order case, the second order interaction effects on price due to a marginal arrival from \( k \neq i \neq j \) are positive. However, as the degree of the denominator terms in equation (10) are higher than that of equation (9), the second order interaction effects \( \frac{\partial^2 p_j^*}{\partial \lambda_y \partial \lambda_{sj}} \) on price are less sensitive to marginal arrivals than the first order effects \( \frac{\partial p_j^*}{\partial \lambda_y} \).

**Lemma 1 (The Marginal Effect on Job Execution Price Lemma).** The marginal effect on price for executing jobs at any location \( j \) due to the marginal arrival of a job from any neighbor location is always positive.

**Proof.** By definition \( \mu_j > \sum_i \lambda_{ij}^* \) since a violation of this condition would lead to infinite delays for all jobs. As a result, \( \left( \mu_j - \sum_i \lambda_{ij}^* \right)^n > 0 \), \( \forall n \), where \( n \) is any integer. Since the denominators in equations (8), (9), and (10) are always positive, \( \frac{\partial p_j^*}{\partial \lambda_{ij}} \) and \( \frac{\partial^2 p_j^*}{\partial \lambda_y \partial \lambda_{sj}} > 0 \).

We ignore interaction effects beyond the second order as there is a negligible effect of marginal job arrivals on marginal price in the higher order expressions. This is because the denominator terms in the higher order expressions are raised to powers greater than four and that results in small fractions. Moreover, as the first and second order expressions consider the pair-wise interactions of all peer neighbors, they can be used to model the marginal effect of a job arrival at any location from all of its neighbors.
We illustrate the aggregate, first, and second order interaction effects due to job arrivals graphically in Figure 3. In Figure 3(a), the optimal price function $p_j^*$ is plotted against effective job arrival rates $\sum_i \lambda_{ij}^*$, while keeping $\mu_j$ constant. We observe that there is an exponential increase in price as total arrival rates approach capacity. For a range of total arrival rates from 0 to 30, the price of job execution at the aggregate level rapidly increases by approximately 0.16 units. The non-linearity of the curve is due to the exponential increase in waiting times for jobs as the effective job arrival rate nears total capacity. In Figure 3(b), the dotted line plots $\frac{\partial p_j^*}{\partial \lambda_{ij}}$ against $\lambda_{ij}$, while keeping $\mu_j$ and $\lambda_{ij}$ constant. We observe here as well that there is an exponential increase in price as $\sum_i \lambda_{ij}^*$ approaches $\mu_j$. However, the rate of increase is less than that of the aggregate price function $p_j^*$. For the same range of job arrivals from 0 to 30, the unit price for marginal job execution increases by about 0.009. In Figure 3(c), the boxed line plots $\frac{\partial^2 p_j^*}{\partial \lambda_{ij} \partial \lambda_{ij}}$ against $\lambda_{ij}$, while keeping $\mu_j$ and $\lambda_{ij}$ constant. There is also a non-linear increase in change in marginal price due to the second order interaction effects when the effective arrival rate approaches capacity. However, as can be observed from the graphs, the second order effects are less sensitive to marginal arrivals compared to the first order effects. In this case, for a 30 units increase in job arrival rates, the execution price increases by only about 0.0012 units.

These plots demonstrate that, since we are considering aggregate effects, $p_j^*$ increases more rapidly with arrival rates than the marginal first and second order effects. In effect, prices are most sensitive to job arrivals at an aggregate level, followed by the first order, and lastly the second order marginal arrivals. This highlights the difference between the aggregate and marginal effects of job arrivals on the optimal price. Therefore, optimal pricing strategies in a P2P network are affected by the structure of neighbors to any location. In addition, the sensitivity of prices to marginal arrivals may be determined based on the network structure.
Note that our expression for the optimal price $p^*_j$ (from equation (4)) is similar to the Mendelson and Whang (1990) result, where the single-server / uniform-job-size model of Mendelson (1985) is extended to include multiple priorities for jobs. This is because our multiple-server setup introduces an additional level of complexity to the basic Mendelson (1985) model, similar to the one introduced by the multiple-priority model of Mendelson and Whang (1990). However, both these earlier models (Mendelson 1985, Mendelson and Whang 1990) ignore higher-order interaction effects because their settings involve only a single server. In our model there are multiple server locations. Therefore, the comparative statics analysis of the first and second order interaction effects $\partial p^*_j / \partial \lambda_{ij}$ and $\partial^2 p^*_j / \partial \lambda_{ij} \partial \lambda_{ij}$ capture the problem characteristics. Our general multiple-server model reduces to the Mendelson (1985) basic single-server model as a special case if we replace $\Gamma_j = \sum_i \lambda_{ij}$ by $\lambda$.

3.2 Maximum Permissible Server Queue Length Extension

We now consider a case where every peer in the network has a maximum permissible queue length constraint. This would correspond to having a quality of service (QoS) guarantee since user delays would now be limited by a specific queue length. This also allows us to model the case where any server has the ability to ‘outsource’ a job to a third party when the utilization level at that server reaches a certain point. The advantage of this model extension is that businesses, or a P2P network of individual nodes in an organization, can utilize it for offering a guaranteed level of service to users. To illustrate this with the earlier base case example, IT firms such as IBM or Akamai can not only offer services but they can also specify an upper limit on the waiting time that customers would experience. The same is true of independent servers in a firm that may choose to form a P2P network only if they are assured of a desired level of service.

Now, the objective of every peer is to maximize its net value, subject to the queue length being no more than a specific exogenous value, say, $q$. Therefore, this constrained optimization problem is as follows:
The Constrained Queue Length Problem

\[
\max_{\Gamma_j} \left\{ \Lambda \left( \Gamma_j \right) - v L_j \left( \Gamma_j \right) \right\}
\]  

The Constrained Queue Length Problem  \hspace{1cm} (11)

subject to \( q - L_j \geq 0 \).  \hspace{1cm} (12)

Note that as long as \( q \) is no more than the queue length corresponding to the optimal solution of equation (3), constraint (12) will be binding. The constrained queue length problem can be solved using a Lagrangean relaxation method, where we relax constraint (12). We use \( \beta_j \) as the Lagrange multiplier and add the relaxed constraint to the objective function. The relaxed problem is as follows:

\[
\max_{\Gamma_j, \beta_j} \left\{ \Lambda \left( \Gamma_j \right) - v L_j \left( \Gamma_j \right) + \beta_j \left( q - L_j \left( \Gamma_j \right) \right) \right\}
\]  

The Lagrangean Relaxation Restated Problem  \hspace{1cm} (13)

subject to \( \beta_j \geq 0 \).  \hspace{1cm} (14)

This restated problem can be solved analytically for observing the effects of queue length constraints on optimal service price.

**Proposition 2 (The Constrained Job Queue Length Job Execution Price Proposition).** When there is a constraint on the maximum permissible job queue length at any location \( j \), the optimal price \( p^*_j \) for executing a job is given by:

\[
p^*_j = \left( v + \beta_j \right) \left[ \Gamma_j^* \left( \partial W_j / \partial \Gamma_j^* \right) \right] + \beta_j W_j \left( \Gamma_j \right)
\]  

(15)

where \( \Gamma_j^* \leq q / W_j \left( \Gamma_j \right) \).  \hspace{1cm} (16)

**Proof.** The FOC with respect to \( \Gamma_j \) for maximizing the Lagrangian relaxation restated problem is as follows:

\[
\Lambda' \left( \Gamma_j \right) - v \left[ \Gamma_j \left( \partial W_j / \partial \Gamma_j^* \right) + W_j \left( \Gamma_j \right) \right] - \beta_j \left[ \Gamma_j \left( \partial W_j / \partial \Gamma_j^* \right) + W_j \left( \Gamma_j \right) \right] = 0.
\]  

(17)

Substituting for \( \Lambda' \left( \Gamma_j \right) = p_j + v W_j \left( \Gamma_j \right) \) into (17) and rearranging terms, we obtain equation (15).

Given that constraint (16) is binding, the FOC with respect to \( \beta_j \) is:
Please cite this article in press as: Kumar, C., et al. A mechanism for pricing and resource allocation in peer-to-peer networks. Electronic Commerce Research and Applications (2010), doi:10.1016/j.elerap.2010.07.004

\[ q - L_j \left( \Gamma_j \right) \geq 0. \]  

(18)

Substituting for \( L_j = \Gamma_j W_j \left( \Gamma_j \right) \) into (18) and rearranging terms, we obtain equation (16).

The present form of the optimal price expression (15) for the constrained queue length case may be compared with the base case expression (4). Since \( \beta_j \geq 0 \) (from (14)), the \( \left( v + \beta_j \right) \) and \( \beta_j W_j \left( \Gamma_j \right) \) terms in (15) represent a job execution price increase compared to the base case (4). Users now have reduced waiting times, where \( W_j \left( \Gamma_j \right) \leq q/\Gamma_j^* \) (from (16)), due to queue length constraints. Consequently, they can be charged a greater unit price for executing jobs. Therefore, a constraint on the maximum permissible queue length reduces delays and improves QoS by increasing the penalty due to per unit waiting time cost \( \left( v + \beta_j \right) \) as well as the overall waiting cost for every job. For this result, we assume that \( q \) is such that the queue length constraint is binding. If this is not the case, then the model reduces to the base case version where there is no constraint on the queue length at any server location.

Note that if \( \Gamma_j W_j \left( \Gamma_j \right) > q \), then price \( p_j^* = \infty \) to ensure that no more jobs are executed at location \( j \).

As in the model base case, the optimal price expression may be represented in terms of the capacity \( \left( \mu_j \right) \) and job arrival rate \( \left( \Gamma_j \right) \) at \( j \). Substituting equations (6) and (7) into (15) and (16) and rearranging terms we obtain:

\[ p_j^* = \begin{cases} 
\left( v + \beta_j \right) \left[ 1/\left( \mu_j - \Gamma_j^* \right) \right] + \beta_j \left[ 1/\left( \mu_j - \Gamma_j^* \right) \right] & \text{if } \Gamma_j W_j \left( \Gamma_j \right) \leq q, \text{ and} \\
\infty & \text{if } \Gamma_j W_j \left( \Gamma_j \right) > q,
\end{cases} \]

(19)

where \( \Gamma_j^* \leq \left[ q/(q+1) \right] \mu_j \).

From equations (19) and (20), we infer that while \( \Gamma_j W_j \left( \Gamma_j \right) \leq q \), the optimal price \( p_j^* \) for executing a job increases exponentially as \( \Gamma_j^* = \sum_i \lambda_i^* \) approaches \( \mu_j \). However, \( p_j^* \) is more sensitive to changes in
\( \Gamma_j^* \) compared to the unconstrained base case price (equation (8)) due to the increased effect of \( \beta_j \geq 0 \).

Another key difference is that due to the queue length constraint, \( p_j^* = \infty \) when \( \Gamma_j W_j(\Gamma_j) > q \).

Thus, we see that we can improve the QoS by reducing the overall waiting time when we impose an exogenous constraint on the maximum permissible queue length \( q \) on server locations. This is due to increased penalty for waiting time costs for jobs. Any desired level of QoS may be achieved by varying \( q \). Using this, peers in the P2P network can compare the benefit of improving QoS for different values of \( q \) versus the increased cost of outsourcing jobs that are not executed at server locations due to the queue length constraints.

### 3.3 Strategic Interaction Among Peers

We derive the optimal transfer price for jobs being executed at different nodes within a firm in our base case model and extension. Since the firm is interested in improving overall performance, a net value maximization objective for peers is appropriate in this scenario. A different area of study is to consider a network where individual peers are behaving strategically. Here the P2P network may be made up of individual users that are not part of an organization and they can behave selfishly. The peers may use shared resources without themselves contributing as they are not necessarily concerned about network performance. Examples of these include the popular file sharing networks such as Kazaa, Gnutella, the first version of Napster, etc. Many of these networks have been shut down or modified due to legal issues of distributing copyrighted material (Jones 2006).

Some studies have proposed mechanisms to reduce ‘free-riding’ in such networks where many users contribute little or nothing. Buragohain et al. (2003) develop a differential service where every peer \( i \) is rewarded by other peers in proportion to its contribution \( d_i \). The study characterises the Nash Equilibrium where peer \( i \) determines whether to join the system if its benefit \( b_i \) is greater than or equal to a critical value \( b_c \). Gupta and Somani (2005) also propose a variation of a reputation index for peers based on past and current service record. Feldman et al. (2004) use a prisoners’ dilemma model to capture
the tension between individual and social utility in file sharing networks. They propose a reciprocative decision function where a shared history of user’s actions is maintained in the network. This is used by peers to determine whether to cooperate or defect with others. Koo and Lee (2007) design a protocol for a P2P content distribution system, which is a network type where jobs may be divided into smaller units, where users have incentives to truthfully reveal willingness-to-pay for services. In this mechanism the capacity that every user makes available to others in turn determines the amount of resources they receive. None of these mechanisms were eventually incorporated due to the legal issues involved with file sharing networks. However, as mentioned earlier, P2P based mechanisms are increasingly used by organizations to effectively utilize resources (Ottolenghi 2007).

Our model is a first-cut attempt at developing a P2P mechanism for a firm with jobs arriving at multiple servers using a transfer pricing approach. Later versions may consider P2P networks with individual users, outside specific organizations, that behave strategically without considering the overall network performance. As with past studies on file sharing networks, a reputation mechanism could possibly be used for improving the network performance in this scenario.

4 Numerical Computations

We now perform some numerical computations for our mechanism. They illustrate the effects of job arrivals on prices and load allocations at the individual servers. We consider a simple P2P network structure of three server locations discussed earlier using Figure 2. The job arrival rates at each of the three neighbor locations, referred to as server 1, 2 and 3, respectively, is simulated using a Poisson process. The mean Poisson arrival rates for the three servers are $\lambda_1 = 3$, $\lambda_2 = 4$, and $\lambda_3 = 5$. The total arrivals in the system at any instant are the sum of the exogenous arrivals at individual locations. Table 1 shows 10 instances of the simulated individual arrivals $\lambda_j$ at location $j$ and the corresponding total arrivals. Figure 4 is a histogram representation of the total system arrival for 100 generated instances of
individual arrivals. It demonstrates that total arrivals also follows a Poisson process with a mean arrival rate of 12 which is the sum of the individual server mean arrival rates.

At equilibrium, the jobs are allocated to servers such that their loads are as close to equal as possible. The detailed proof that this is indeed optimal for the P2P network is demonstrated in Proposition 3 below. This means that if there are 9 total jobs in the system, 3 jobs are allocated to each of the 3 servers. In this case the unit price for executing jobs at each server location is identical. However, if total jobs are not a perfect multiple of the number of nodes in the P2P network, then the extra jobs are randomly allocated to some servers. In this case, there is an imbalance in the prices at the different server locations. Table 2 shows the job arrivals, job allocations, and prices for executing jobs at servers for 10 instances. The job allocation algorithm is as follows: INT and MOD refer to Integer and Modulo functions, respectively. Each server receives INT (total jobs/3) number of jobs. If (total jobs MOD 3) is 1 or 2, then server 2 receives an additional job. If (total jobs MOD 3) is 2 then server 3 also receives an additional job. Using this algorithm, we can calculate job allocations $\Gamma_j = \sum_{i} \lambda_{ij}$ for any server $j$, where $i$ is a neighbor location to $j$.

We earlier derived the optimal price expression (8) in terms of the capacity $\mu_j$ and job allocation $\Gamma_j$ at any $j$. We can now use expression (8) to calculate numerical values for the price $p_j$ for executing jobs at each server $j$. Let server capacity $\mu_j = 14$ for all $j$ and unit waiting time cost $v = 5$. Note that server capacities are kept constant in order to demonstrate the effect of job arrivals alone to allocations and pricing. Using these values, the price $p_j$ for executing jobs corresponding to $\Gamma_j$ is shown in Table 2. Since server 2 and server 1 are allocated the extra jobs after equal distribution, they have higher prices in some instances. By regulating prices according to the level of job congestion, the server loads will be balanced out in the long run. Figure 5 shows how prices for executing jobs at each server location are affected by total arrivals. We observe that every time the total arrivals are a perfect multiple of 3, the
prices are identical. Otherwise, server 2 and server 1 prices may be higher. This leads us to the following general proposition about prices and job arrivals.

**Proposition 3 (The Equal Job Executions and Prices based on Total Number of Nodes and Jobs Proposition).** For direct neighbors in a P2P network the number of jobs executed at any node \( j \) is \[ M = \sum_j \lambda_j / N, \] where \( N \) is the total number of direct neighbors, and the price \( p^*_j \) for executing a job is the same at any neighbor node if and only if the ratio \( M \) is an integer.

**Corollary 1 (The Unequal Job Executions and Prices based on Total Number of Nodes and Jobs Corollary).** If the ratio \( M = \sum_j \lambda_j / N \) is not an Integer, then there is a difference in price \( p^*_j \) for executing jobs at different locations \( j \) in a P2P network.

The proofs of Proposition 3 and Corollary 1 are provided in the Appendix.

5 **Discussion and Conclusions**

In this study, we design a pricing and allocation mechanism for a P2P network that allows users in a firm to effectively share their computing resources. This mechanism allows tasks (incoming jobs) to be allocated to resources (participating peers) in an organization in a decentralized manner. The base case of our P2P mechanism derives the optimal transfer price that maximizes the net value for the peers in the network. The optimal price is equal to the externality cost, which is the expected delay cost per unit time inflicted on the rest of the system by an infinitesimal increase in job flow. Therefore, the optimal price for executing any job at a peer location is essentially equal to the marginal delay cost it imposes on all current jobs at that node. With this pricing scheme no individual user has an incentive to overutilize shared resources, thereby avoiding the ‘Tragedy of the Commons’ for a P2P network.

Our model builds on the Mendelson (1985) study that was one of the first to look at the control and management issues related to a computer system. The transfer pricing model of Mendelson (1985) was later extended by Mendelson and Whang (1990) to include multiple priorities and heterogeneous
sizes. However, these earlier studies considered the case of pricing for a single server with multiple jobs arrivals. In this paper, we model a multiple server environment of a P2P network where the users may share their computing resources. The original Mendelson (1985) model thus becomes a special single-server case of our general model. Our basic model is extended to incorporate situations where the peers have queue length constraints. The optimal price in the queue length constraint extension, which may be used for providing QoS guarantees to users, is derived using a Lagrangean relaxation method. We then perform numerical computations that illustrate the effects of job arrivals on prices and load allocations at the individual servers.

P2P networks have properties that make them an interesting research area. For example, every individual has both a positive and a negative externality effect on other users. In addition, P2P networks are inherently scalable, have no single point of failure, and possess self-administration capabilities. Some of the largest IT companies such as Microsoft, HP, and IBM have invested significant resources in P2P applications (Ottolenghi 2007). P2P based technologies have enormous potential for helping organizations manage their computer resources effectively. For instance, using our mechanism organizations can decide how to price and allocate the jobs such that, given the objective of the peers, the outcome is optimal for the network. In addition, they can manage the network to provide a guaranteed level of QoS to users. Given these properties, a P2P based mechanism provides a useful methodology for organizations to better utilize and share their computing resources. Therefore, we believe that this study contributes to an important area for IS research.

There are a number of interesting areas for future research. Our model derives the prices that are optimal for the individual peer locations in a P2P network. The model can be extended to develop an optimal pricing mechanism while considering all the nodes together in a P2P network, which is equivalent to a centralized pricing scheme. In addition, there may be some interaction terms and higher order derivatives in the numerical expansion of the waiting time externality expression $\frac{\partial W_j}{\partial \Gamma_j}$ in terms of $\lambda_j$. We plan to investigate approximation methods, such as Newton’s method, for representing
the non-linear externality term numerically. The incentive compatible nature of our mechanism can be addressed from a game-theoretic perspective. We aim to characterize the Nash Equilibrium where each individual job decides whether to join a particular node, given the prices. Besides the net value maximization peers’ objective utilized in this study, one could consider other objectives such as profit maximization and delay cost minimization. One could also investigate the incentives under which peers may choose these alternative objectives. Further, one could model the case where the jobs might have different sizes. In our queue length constraint model extension, an area of future research could be to compare the benefit of improving QoS for different values of $q$ versus the increased cost of outsourcing jobs that are not executed at server locations due to constraints. Additionally, it may be worthwhile to compare the performance of our decentralized model with that of a centralized model. In the latter scenario, a job arriving at any node may be allocated to any destination by a central administrator. Various network structures (such as ARPAnet, USAnet, etc.) could be used for this comparison. Finally, it could also be interesting to consider a variation of a P2P network composed of individual users, not part of an organization, who are behaving strategically. Alternative approaches beyond transfer pricing have to be developed for such a scenario.

To the best of our knowledge, our research is among the first to develop a pricing and allocation mechanism for a multi server P2P network setting where users in a firm can share their computing resources. We believe that our mechanism provides a useful methodology for organizations to better utilize and share their computing resources, based on the analysis and conclusions presented in this study.

References


Bhattacharjee, S., Gopal, R. D., Lertwachara, K., Marsden, J. R., Telang, R. The Effect of Digital Sharing


Please cite this article in press as: Kumar, C., et al. A mechanism for pricing and resource allocation in peer-to-peer networks. Electronic Commerce Research and Applications (2010), doi:10.1016/j.elerap.2010.07.004


Appendix

Figure 1. A P2P network for sharing computing resources

Figure 2. A simple example of a P2P Network with three server locations
(a) Aggregate Price Function

(b) First Order Interaction Effect

(c) Second Order Interaction Effect

Figure 3. Aggregate and Marginal job arrival effects
Table 1. Individual and total arrivals

<table>
<thead>
<tr>
<th>Job Arrivals $\lambda_j$</th>
<th>Total Arrivals</th>
</tr>
</thead>
<tbody>
<tr>
<td>server 1</td>
<td>server 2</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
</tr>
</tbody>
</table>

Figure 4. Histogram of total job arrivals
Table 2. Job arrivals, allocations, and prices

<table>
<thead>
<tr>
<th>Job Arrivals $\lambda_j$</th>
<th>Job Allocations $\Gamma_j$</th>
<th>Prices $p_j$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>server 1</td>
<td>server 2</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 5. Price for executing jobs at servers and total arrivals
Proof of Proposition 3. If the total job arrivals across all direct neighbors, $\sum_j \lambda_j$, is a multiple of $N$, then $M = \sum_j \lambda_j / N$ is an integer. In this case, there are only two possible cases as to how the jobs may be allocated, namely, Case 1: The job executions are not equally allocated at $M$ jobs per node $j$, and Case 2: The jobs are equally allocated at $M$ jobs for every $j$. We now compare the prices, and corresponding effect on the peers’ objective function, for executing jobs for both cases.

Case 1: Let a node $j'$ have job allocation $\Gamma_j' > M$ jobs. Since the total number of jobs is constant at $\sum_j \lambda_j$, this necessarily means that another node $j''$ has $\Gamma_j'' < M$ jobs. Here $j' \neq j'' \neq j$. Using expression (4) to arrive at execution prices at $j'$ and $j''$ for these job allocations, we obtain:

$$p_j = v\Gamma_j \left( \partial W_j / \partial \Gamma_j \right) \quad \text{and} \quad p_j = v\Gamma_j \left( \partial W_j / \partial \Gamma_j \right).$$

Since $\Gamma_j \left( \partial W_j / \partial \Gamma_j \right) > \Gamma_j \left( \partial W_j / \partial \Gamma_j \right)$, we have $p_j > p_j'$ and there is a price differential for executing jobs at the two nodes. Again using (4), let $p_j^* = v\Gamma_j^* \left( \partial W_j / \partial \Gamma_j^* \right)$ be the price for executing jobs at any $j$ if it had exactly $\Gamma_j^* = M$ allocated jobs. We know from our earlier discussion using expression (7) that $\partial W_j / \partial \Gamma_j = 1 / \left( \mu - \Gamma_j \right)^2$, the incremental delay added to job waiting time $W_j$ at $j$ as a result of an incremental arrival $\Gamma_j$, increases non-linearly with $\Gamma_j$ for a given $\mu_j = \mu$. We now substitute in (7) for $\partial W_j / \partial \Gamma_j^*$, $\partial W_j / \partial \Gamma_j'$, and $\partial W_j / \partial \Gamma_j^*$ corresponding to allocations $\Gamma_j > M$, $\Gamma_j < M$, and $\Gamma_j^* = M$, respectively. Let $\Gamma_j = M + \Delta_j$, $\Gamma_j = M - \Delta_j$, and $\mu_j = \mu$. Here $\Delta_j$ and $\Delta_j'$ are the increased and reduced allocations at $j'$ and $j''$, respectively, compared to $M$ jobs at each node. Using these, we arrive at the following:

$$\partial W_j / \partial \Gamma_j^* - \partial W_j / \partial \Gamma_j^* = 1 / \left( \mu - (M + \Delta_j) \right)^2 - 1 / \left( \mu - M \right)^2$$

and

$$\partial W_j / \partial \Gamma_j^* - \partial W_j / \partial \Gamma_j^* = 1 / \left( \mu - (M - \Delta_j) \right)^2 - 1 / \left( \mu - M \right)^2$$
We compare $\partial W_j / \partial \Gamma_j - \partial W_j / \partial \Gamma_j^*$ and $\partial W_j / \partial \Gamma_j^* - \partial W_j / \partial \Gamma_j$ by subtracting (23) from (22) to obtain:

$$
\left( \partial W_j / \partial \Gamma_j - \partial W_j / \partial \Gamma_j^* \right) - \left( \partial W_j / \partial \Gamma_j^* - \partial W_j / \partial \Gamma_j \right) = \\
1/\left( \mu - (M + \Delta_j) \right)^2 + 1/\left( \mu - (M - \Delta_j) \right)^2.
$$

(24)

There are three possible cases while considering $\Delta_j$ and $\Delta_j^*$: $\Delta_j > \Delta_j^*$, $\Delta_j = \Delta_j^*$, and $\Delta_j < \Delta_j^*$. However, $\Delta_j < \Delta_j^*$ is identical to $\Delta_j > \Delta_j^*$ with $j'$ and $j''$ in reverse order. Therefore, we only need to consider the cases of $\Delta_j > \Delta_j^*$ and $\Delta_j = \Delta_j^*$. In both these cases, expression (24) is positive as $\left( \mu - (M + \Delta_j) \right)^2 > 0$ and $\left( \mu - (M - \Delta_j) \right)^2 > 0$. Therefore, we obtain the following condition:

$$
\partial W_j / \partial \Gamma_j - \partial W_j / \partial \Gamma_j^* > \partial W_j / \partial \Gamma_j^* - \partial W_j / \partial \Gamma_j.
$$

(25)

Now, since $\partial W_j / \partial \Gamma_j > \partial W_j / \partial \Gamma_j^*$, the following is also true:

$$
\Delta_j \partial W_j / \partial \Gamma_j > \Delta_j \partial W_j / \partial \Gamma_j^*.
$$

(26)

From (25) and (26), we obtain:

$$
M \partial W_j / \partial \Gamma_j - M \partial W_j / \partial \Gamma_j^* + \Delta_j \partial W_j / \partial \Gamma_j > \\
M \partial W_j / \partial \Gamma_j^* - M \partial W_j / \partial \Gamma_j + \Delta_j \partial W_j / \partial \Gamma_j.
$$

(27)

Substituting for $\Gamma_j^*$, $\Gamma_j$, and $\Gamma_j^*$ in (27) and rearranging, we arrive at the following condition:

$$
\Gamma_j \partial W_j / \partial \Gamma_j - \Gamma_j^* \partial W_j / \partial \Gamma_j^* > \Gamma_j^* \partial W_j / \partial \Gamma_j^* - \Gamma_j \partial W_j / \partial \Gamma_j.
$$

(28)

Substituting for (21) into (28), we obtain:

$$
p_j^* - p_j = v(\Gamma_j \partial W_j / \partial \Gamma_j - \Gamma_j^* \partial W_j / \partial \Gamma_j^*) > \\
p_j^* - p_j = v(\Gamma_j^* \partial W_j / \partial \Gamma_j^* - \Gamma_j \partial W_j / \partial \Gamma_j^*).
$$

(29)
From expression (29), we observe that every job executed at $j'$ has a higher increase in prices $p_j - p_j^*$ due to increased allocations $\Gamma_j - \Gamma_j^*$ than the corresponding reduction in prices $p_j - p_j^*$ at $j''$ due to reduced allocations $\Gamma_j^* - \Gamma_j$. This can also be shown by rearranging $p_j - p_j^* > p_j^* - p_j$ from (29) and we obtain:

$$p_j + p_j^* > 2p_j^*. \tag{30}$$

From (30), we know that the total price of executing jobs $p_j + p_j^*$ at unequal $\Gamma_j$ and $\Gamma_j^*$ allocations is greater than the price $2p_j^*$ if the nodes had equal $\Gamma_j^*$ job allocations. This, in turn, indicates that the net value maximization objective function of peers is not maximized in the unequal allocations scenario, discussed as follows.

From Proposition 1, we know that the peer locations have a net value maximization objective function. We now consider the effect of prices for executing jobs, and job allocations, at nodes $j'$ and $j''$ together on this peers’ objective. Let $\Lambda(\Gamma_j')$, $\Lambda(\Gamma_j)$, and $\Lambda(\Gamma_j^*)$, represent the gross value of jobs gained by systems users when the allocated jobs are $\Gamma_j' > M$, $\Gamma_j' < M$, and $\Gamma_j^* = M$, respectively. We know that $\Lambda(\Gamma_j')$ for any $\Gamma_j$ is a strictly concave value function. Therefore, we arrive at the following condition:

$$\Lambda(\Gamma_j') - \Lambda(\Gamma_j^*) \leq \Lambda(\Gamma_j^*) - \Lambda(\Gamma_j^*). \tag{31}$$

Rearranging (31), we obtain:

$$\Lambda(\Gamma_j') + \Lambda(\Gamma_j^*) \leq 2\Lambda(\Gamma_j^*). \tag{32}$$

Let $W_j'$, $W_j^*$, and $W_j^*$ be the expected waiting times observed by jobs for the corresponding job allocations, $\Gamma_j'$, $\Gamma_j$, and $\Gamma_j^*$, respectively. (For simplicity we use $W_j'$, $W_j^*$, and $W_j^*$ henceforth, which correspond to $W_j(\Gamma_j')$, $W_j(\Gamma_j)$, and $W_j^*(\Gamma_j^*)$, respectively.) We know from expression (6) that the
job waiting time $W_j = \frac{1}{(\mu_j - \Gamma_j)}$ at any $j$ increases non-linearly with higher job allocations $\Gamma_j$ for a given $\mu_j = \mu$. Therefore, the increase in waiting time $W_{j'} - W_j^*$ at $j'$ due to higher allocations $\Gamma_{j'} - \Gamma_j^*$ is greater than the reduction in waiting time $W_j^* - W_j$ at $j''$ due to reduced allocations $\Gamma_j^* - \Gamma_{j''}$. As a result, using the same reasoning as for (28) earlier, we arrive at the following relationship:

$$v\Gamma_j W_{j'} - v\Gamma_j^* W_j^* > v\Gamma_j^* W_j^* - v\Gamma_j W_{j'}.$$

(33)

Rearranging (33), we obtain:

$$v\Gamma_j W_{j'} + v\Gamma_j W_j > 2v\Gamma_j^* W_j^*.$$

(34)

Now, substituting for job allocations $\Gamma_{j'}$ and $\Gamma_j$ into objective function (3) for each node, the total net value for executing jobs at $j'$ and $j''$ together is:

$$\left\{ \Lambda\left(\Gamma_{j'}\right) - v\Gamma_{j'} W_{j'} \right\} + \left\{ \Lambda\left(\Gamma_j\right) - v\Gamma_j W_j \right\}.$$

(35)

From (32) and (34) we arrive at the following:

$$\left\{ \Lambda\left(\Gamma_{j'}\right) - v\Gamma_{j'} W_{j'} \right\} + \left\{ \Lambda\left(\Gamma_j\right) - v\Gamma_j W_j \right\} < 2\left\{ \Lambda\left(\Gamma_j^*\right) - v\Gamma_j^* W_j^* \right\}.$$

(36)

From (36) we see that the net value for executing jobs at nodes $j'$ and $j''$ together is less than the case when both nodes have $\Gamma_j^* = M$ allocated jobs. Therefore, the net value objective of peers is not maximized when nodes $j'$ and $j''$ have unequal job allocations, $\Gamma_{j'} > M$ jobs and $\Gamma_{j''} < M$, respectively.

Case 2: Here, the job executions are equally distributed at $\Gamma_j^* = M$ jobs per node $j$. Since all nodes $j$ have equal allocations, they have an identical price of $p_j^* = v\Gamma_j^* \left( \partial W_j / \partial \Gamma_j^* \right)$ for executing jobs.

The total price for executing jobs at all $N$ direct neighbors in the P2P network is:

$$Np_j^* = Nv\Gamma_j^* \left( \partial W_j / \partial \Gamma_j^* \right).$$

(37)

From (30), we know that with an identical $p_j^*$ price per node $j$, the total price $Np_j^*$ for executing jobs in the P2P network is always less than the case where any nodes have unequal job
allocations. Correspondingly, the net value for executing jobs at every node is \( \{ \Lambda(\Gamma_j^*) - \nu \Gamma_j^* W_j^* \} \) for equal \( \Gamma_j^* = M \) job allocations. Therefore, the total net value of the peers for executing jobs with equal \( \Gamma_j^* = M \) allocations is:

\[
N \{ \Lambda(\Gamma_j^*) - \nu \Gamma_j^* W_j^* \}. 
\]  

(38)

From (36), we know that \( N \{ \Lambda(\Gamma_j^*) - \nu \Gamma_j^* W_j^* \} \) is the maximum net value possible for executing jobs in the P2P network compared to unequal allocations at any nodes. Consequently, equal job allocations \( \Gamma_j^* = M \) at every \( j \) maximizes the total net value of the peers in the network. Therefore, the solution in Case 1 is inferior to the solution in Case 2. Thus, if jobs are allocated equally at \( \Gamma_j^* = M \) jobs per node \( j \), the corresponding identical price of \( p_j^* = \nu \Gamma_j^* \left( \frac{\partial W_j}{\partial \Gamma_j^*} \right) \) at any \( j \) is optimal for the P2P network when the total job arrival \( \sum_j \lambda_j \) is a multiple of number of direct neighbors \( N \).

Proof of Corollary 1. If \( \sum_j \lambda_j \) MOD \( N \neq 0 \), then \( M \) is obviously non-integral. Individual jobs cannot be broken up and allocated fractionally to nodes. Then, INT(\( M \)) jobs are equally distributed among all nodes \( j \), and \( \sum_j \lambda_j \) MOD \( N \) jobs are arbitrarily allocated to some nodes. Therefore, a price differential exists due to different job allocations at different peer locations.
Table of Notations

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>$j$</td>
<td>server location</td>
<td></td>
</tr>
<tr>
<td>$i, k$</td>
<td>‘direct neighbor’ locations of $j$</td>
<td>neighbor nodes have direct network links between themselves</td>
</tr>
<tr>
<td>$\lambda_j$</td>
<td>exogenous arrival rate of a job at $j$</td>
<td></td>
</tr>
<tr>
<td>$\lambda_{ij}$</td>
<td>arrival rate of a job originating at $i$ to be executed at $j$</td>
<td>also referred as allocation of jobs from $i$ to $j$</td>
</tr>
<tr>
<td>$\Gamma_j = \sum_i \lambda_{ij}^\ast$</td>
<td>total / optimal total arrival rate of jobs at server $j$</td>
<td>also referred as total allocation of jobs at $j$</td>
</tr>
<tr>
<td>$\Lambda\left(\Gamma_j\right)$</td>
<td>gross value gained by system users at $j$ when the arrival rate is $\Gamma_j$</td>
<td></td>
</tr>
<tr>
<td>$W_j\left(\Gamma_j\right)$ or $W_j$</td>
<td>expected time a job spends at $j$ when the arrival rate is $\Gamma_j$</td>
<td>$W_j$ is used instead of $W_j\left(\Gamma_j\right)$ for simplicity</td>
</tr>
<tr>
<td>$L_j\left(\Gamma_j\right)$ or $L_j$</td>
<td>expected job queue length at $j$ when the arrival rate is $\Gamma_j$</td>
<td>$L_j$ is used instead of $L_j\left(\Gamma_j\right)$ for simplicity</td>
</tr>
<tr>
<td>$v$</td>
<td>delay cost per-unit time for any job</td>
<td></td>
</tr>
<tr>
<td>$p_j / p_j^\ast$</td>
<td>price / optimal price for executing a job at node $j$</td>
<td></td>
</tr>
<tr>
<td>$\partial W_j / \partial \Gamma_j$ or $\partial W_j / \partial \Gamma_j^\ast$</td>
<td>incremental delay added to $W_j$ due to an incremental arrival $\Gamma_j$ / $\Gamma_j^\ast$</td>
<td></td>
</tr>
<tr>
<td>$\partial L_j / \partial \Gamma_j$</td>
<td>incremental increase in queue length $L_j$ due to an incremental arrival $\Gamma_j$</td>
<td></td>
</tr>
<tr>
<td>$\mu_j$</td>
<td>job processing capacity at $j$</td>
<td></td>
</tr>
<tr>
<td>$\partial p_j^\ast / \partial \lambda_{ij}$</td>
<td>first order effect, marginal change in price at $j$ due to marginal arrival $\lambda_{ij}$ from $i \neq j$</td>
<td></td>
</tr>
<tr>
<td>$\partial^2 p_j^\ast / \partial \lambda_{ij} \partial \lambda_{kj}$</td>
<td>second order interaction effect, change in marginal price at $j$ due to marginal arrivals $\lambda_{ij}$ and $\lambda_{kj}$ from neighbor locations $k \neq i \neq j$</td>
<td></td>
</tr>
<tr>
<td>$q$</td>
<td>maximum permissible queue length</td>
<td>can be used for providing QoS guarantees</td>
</tr>
</tbody>
</table>
\[ M = \sum_j \frac{\hat{\lambda}_j}{N} \] number of jobs executed at any node \( j \)

\[ N \] total number of direct neighbors in the P2P network

\( j' \) a node that has job allocations \( \Gamma_{j'} > M \) jobs

\( j'' \) another node that has job allocations \( \Gamma_{j''} < M \) jobs

\[ p_{j'} = \nu \Gamma_{j'} \left( \frac{\partial W_{j'}}{\partial \Gamma_{j'}} \right) \] job execution price at \( j' \)

\[ p_{j''} = \nu \Gamma_{j''} \left( \frac{\partial W_{j''}}{\partial \Gamma_{j''}} \right) \] job execution price at \( j'' \)

\[ \Delta_{j'} \] increased job allocations at \( j' \) where \( \Gamma_{j'} = M + \Delta_{j'} \)

\[ \Delta_{j''} \] reduced job allocations at \( j'' \) where \( \Gamma_{j''} = M - \Delta_{j''} \)

\[ \Lambda (\Gamma_{j'}) \text{ and } W_{j'} \] gross value of executing jobs and expected waiting time at \( j' \)

\[ \Lambda (\Gamma_{j''}) \text{ and } W_{j''} \] gross value of executing jobs and expected waiting time at \( j'' \)

\[ \Lambda (\Gamma_{j}) \text{ and } W_{j}^{*} \] gross value of executing jobs and expected waiting time when jobs are equally allocated at \( \Gamma_{j}^{*} = M \) jobs per node \( j \) and the nodes have an identical execution price \( p_{j}^{*} = \nu \Gamma_{j}^{*} \left( \frac{\partial W_{j}}{\partial \Gamma_{j}^{*}} \right) \)

\[ \beta_j \] Lagrange multiplier for solving maximum permissible server queue length extension problem

notations used in numerical computations section for Proposition 3, and Corollary 1, regarding job executions and prices based on total number of nodes and jobs