Cooperative and Non-cooperative Resource Sharing in Networks: A Delay Perspective

Tara Javidi

Abstract

From multi-description/multi-path routing for multi-media applications to content distribution in P2P networks, to community networking, many forms of resource sharing have been proposed to improve network performance. From the perspective of any single user, when ignoring the interaction among users, all such schemes reduce to various forms of providing parallelism and, hence, increased throughput. Focusing on parallelism is by no means sufficient as it ignores the existence of many users with potentially similar strategies. In this paper, we focus instead on the delay performance of a multi-user system where resources are shared.

We illustrate the benefit, in an average delay sense, of resource sharing among many (potentially strategic) users via a multi-queue multi-server problem. We use a fork-join queuing model to provide analytical results in a special case of homogeneous users and servers. Our proposed model is simplistic, and yet, it does capture the trade-off between parallelism, traffic load increase, and reassembly/synchronization delay to a large extent. Furthermore, we prove the robustness of a certain locally optimal strategy to non-cooperation in a Nash equilibrium sense.

I. Introduction

Scheduling and resource allocation have become important topics of interest when considering network performance. With the growth of networking technologies, cooperation has received much attention as a means of enabling efficient allocation of resources across networks. Due to the size and decentralized nature of networks today, many applications rely heavily upon resource pooling and sharing to overcome the inherent limitations of the system. Among such applications, one can point to multi-path routing for multi-media applications, content distribution in peer-to-peer (P2P) networks, and community networking.

Consider the issue of content distribution in P2P networks as a technology solution to facilitate remote access to files which reside in individuals' "home" machines. Usually though, the remote access is negatively affected by the asymmetry in upload/download speeds. To overcome this limitation caused by low upload speeds, content distribution solutions rely on cooperation among many users such that the content of one's files is distributed over many home machines allowing the remote access a parallel upload of these distributed pieces. It is then intuitive to find an increased peak rate. This problem has been introduced previously in [1], where strategy-proof sharing policies have been studied and analyzed. Furthermore, it has been shown that such policies result in a fair sharing of the "additional unused" bandwidth of the network. Our work complements these results by focusing on the delay improvements that result from similar cooperation strategies. In particular, we provide a simple queueing theoretic approach to capture the benefit of content distribution with respect to delay, even in scenarios where throughput increase is not feasible.

Delay improvements can be similarly obtained in the context of multi-description coding and multi-path routing [8], [17] schemes which have been proposed to improve quality of service and delay profile for multi-media applications, especially video streaming. While many authors have suggested disjoint (and simultaneous) multi-path routing of the descriptions of a file to improve session delay, most studies have focused on the impact of multi-path routing on one particular user, while assuming a fixed network model [8], [17], [18], [23]. Here, however, we are more interested in understanding the impact of multi-path routing from a network perspective where many users follow similar strategies of multi-path routing. To isolate the impact of statistical multiplexing and avoid the more complicated and topology dependent affects of resource sharing, e.g., alleviating bottle-necks, we focus on a simple fork-join queuing model. Our analysis can also be used to examine the benefits of community networking and access point sharing. Many new schemes of access point

T. Javidi is with the Department of Electrical and Computer Engineering at the University of California, San Diego. This research was in part funded by Center for Networked Systems research grants CNS0509 and CNS0512, Center for Wireless Communications (CWC) projects funded by ViaSat and LGE. Parts of this work was presented at 44th Allerton Conference, 2006.

sharing have started to appear as popular solutions across communities (for one example, see [12]). In such models, each user has more than one wireless interface card and can simultaneously transmit to many access points. We believe that by focusing on an intuitive and simple model/solution, the contribution of our work is in leading the first steps towards delay analysis in the above systems.

In this paper we present a multi-queue multi-server model with stochastic job distributions to capture the salient features of the above set of problems. In our model, which is known as a fork-join queue, the newly arrived jobs are divided into pieces and sent to many queues and servers in order to take advantage of parallelism. In this queueing model, the performance measure of interest is the average response time of a given job, where "job" corresponds to sessions, multi-media files, etc. In a system with many (potentially) strategic users, no throughput increase is possible. Therefore, our result isolates statistical multiplexing as an important mechanism to provide delay improvements.

Here we tackle the issue of delay performance of sharing policies from three distinct (but related) angles. First we consider an upper bound on the performance of sharing policies by ignoring the cost of sharing (overhead/redundancy) when considering the following two questions:

1. In the absence of backlog information, what is the optimal (with respect to session delay) sharing policy among all policies with a symmetric use of servers?

2. How robust is such an optimal symmetric sharing policy to non-cooperative behavior?

After analyzing the above problems, we look at the possible extensions of the problem when considering the impact of overhead increase. In particular, we introduce two models of sharing cost and analyze their impact asymptotically as the number of servers in the system grows large by answering the last question:

3) In the presence of overhead associated with sharing, how does the performance scale with the number of available servers?

Note that here we do not consider cases where cooperation also provides a load balancing benefit, such as those in [6] or an increase in throughput, such as those usually discussed in a cooperative communications context [15]. This should not be interpreted as neglecting the importance of cooperation in improving throughput; rather, it is our attempt to isolate and quantify the delay improvements which can be brought by designing appropriate cooperation even in scenarios where cooperation does not improve throughput.

The remainder of the paper is organized as follows. Section II is a review of basic concepts and facts in stochastic ordering and queueing theory which will later be used in our model and subsequent analysis. In Section III, we present a simple queuing model which captures some salient features of the problem in the case of homogenous users and servers. In Section III, we discuss prior work related to our queuing model. In Section IV, we demonstrate that a policy which divides and distributes jobs in equal size pieces is optimal among all policies that share servers symmetrically. In Section V, we show that the aforementioned optimal symmetric sharing policy is also a Nash equilibrium for the non-cooperative job distribution game. In other words, we show that this policy not only improves the delay performance of the system for all users, but also exhibits robustness to non-cooperation. In Section VI, first we show that for small enough redundancy/overhead such policy outperforms a non-sharing policy, where each stream of jobs is exclusively assigned to a server. We also provide results on the asymptotic analysis of delay performance in presence of redundancy/overhead. Section VII is the summary and conclusion.

Notation: We close this section with a word on the notation used throughout the paper. Boldfaced characters are used to specify vectors and matrices. We use $[1/n]_{n\times m}$ to denote an $n \times m$ matrix whose elements are all $1/n$; we drop the subscript $n \times m$ when there is no ambiguity with respect to the dimensions of a matrix. We user $\mathbf{1}$ to denote a vector whose elements are all one. We use $\#(S)$ to represent the cardinality of set S and $\mathcal{P}_n$ is the set of all n permutations (again the subscript n is dropped when there is no ambiguity). Finally we follow the conventional game theoretic notation of (α^i, α^{-i}) when noting the strategy profile of user i, α^i along with the collective strategy of all users but i, α^{-i}.

II. Stochastic Orders and Queues: A Review

In this section, we provide a brief review of fundamental definitions, concepts, and relevant results in stochastic ordering and queueing theory. Since we use these notions and definitions in our modeling, as well as analysis, the following section is essential in facilitating the understanding of this paper, particularly by readers not familiar with the topics. Readers knowledgable about the topic can skip this section. Except for Fact 3, and Propositions 1-3, the material in this section can be found in [10], [19], and [20].

Definition 1: An n-variate random vector, $\mathbf{X}$ is called *associated* if

$$E(f(\mathbf{X})g(\mathbf{X})) \geq E\left(f(\mathbf{X})\right) E\left(g(\mathbf{X})\right)$$

for all pairs of monotone non-decreasing mappings $f, g : \mathbb{R}^n \to \mathbb{R}$

Note that associated property can be interpreted as a strong generalization of positive correlation property.

Definition 2: An n-variate random vector is called *exchangeable* if its joint distribution is permutation invariant.

Note that the elements of a vector of exchangeable random variables have common marginals. Two trivial examples of exchangeable random vectors are (X, X) and the vector of i.i.d random variables, (X_1, X_2)

Fact 1 (Proposition B.4 in [19]) Let $X_1, \ldots, X_n$ be exchangeable random variables. Also let $\Phi(\boldsymbol{X}; \boldsymbol{\alpha}) = \phi(w(x_1, a_1), \ldots, w(x_n, a_n))$ where ϕ is symmetric, increasing, and convex (on $\mathbb{R}^n$), and for each fixed z, $w(z, .)$ is convex (on $\mathbb{R}$). With the appropriate measurability,

$$\psi(\boldsymbol{\alpha}) = E\Phi(\boldsymbol{X}; \boldsymbol{\alpha})$$

is symmetric and convex on $\mathbb{R}^n$.

Definition 3: Let $\boldsymbol{X}$ and $\boldsymbol{Y}$ be random vectors with values in $\mathbb{R}^n$. $\mathbf{X}$ is said to be less than $\mathbf{Y}$ with respect to *(usual) stochastic order* (written $\boldsymbol{X} \leq_{st} \boldsymbol{Y}$) if $Ef(\boldsymbol{X}) \leq Ef(\boldsymbol{Y})$ for all bounded increasing functions $f : \mathbb{R}^n \to \mathbb{R}$ for which the expectations exist.

Fact 2 (Theorem 4 in [7]) Consider an exchangeable random vector $(X_1, \ldots, X_n)$ and a vector of independent random variables $(\hat{X}_1, \ldots, \hat{X}_n)$ with the same marginal distributions. Then

$$\max_i X_i \leq_{st} \max_i \hat{X}_i.$$

Definition 4: Let $\boldsymbol{X}$ and $\boldsymbol{Y}$ be random vectors with values in $\mathbb{R}^n$ and with finite expectations. Then $\boldsymbol{X}$ is said to be less than $\boldsymbol{Y}$ with respect to *convex order* (written $\boldsymbol{X} \leq_{cx} \boldsymbol{Y}$) if $Ef(\boldsymbol{X}) \leq Ef(\boldsymbol{Y})$ for all convex functions $f : \mathbb{R}^n \to \mathbb{R}$ such that the expectations exist.

Definition 5: Let $\boldsymbol{X}$ and $\boldsymbol{Y}$ be random vectors with values in $\mathbb{R}^n$ and with finite expectations. Then $\mathbf{X}$ is said to be less than $\mathbf{Y}$ with respect to *increasing convex order* (written $\mathbf{X} \leq_{icx} \mathbf{Y}$) if $Ef(\mathbf{X}) \leq Ef(\mathbf{Y})$ for all increasing convex functions $f : \mathbb{R}^n \to \mathbb{R}$ such that the expectations exist.

Fact 3: If $X \leq_{icx} Y$, random vector Z is independent of both X and Y, and $g : \mathbb{R}^n \to \mathbb{R}^k$ is an increasing convex function, then $g(X + Z) \leq_{icx} g(Y + Z)$.

Proof:

For any increasing convex function f, define $f_X(z) = Ef(g(X + z))$ and $f_Y(z) = Ef(g(Y + z))$, where the expectation is taken with respect to X and Y, respectively (all expectations are assumed to exist). From the definition of increasing convex order and convexity of mapping $u \to f(g(u + z))$ for all real vectors z, we have $f_X(z) \leq f_Y(z)$ for all $\forall z \in \mathbb{R}^n$. Therefore, for any increasing convex function f

$$Ef(g(X + Z)) \leq Ef(g(Y + Z)),$$

where the expectations, now, are also with respect to Z. This implies the increasing convex order and, hence, the assertion of the fact. ■

Fact 4 (Theorem 3.4.2 of [20]) Let X and Y be n-dimensional random vectors with finite expectations.

$$X \leq_{icx} Y \iff \exists \hat{Y} =_{st} Y \text{ s.t. } E(\hat{Y}|X) \geq X \quad a.s. \tag{1}$$

Proposition 1: Let $X_1, \dots, X_n$ be exchangeable random variables and $f_1, \dots, f_n$ measurable real functions. Define the function $\overline{f}$ by

$$\overline{f}(y) = \frac{1}{n} \sum_{i=1}^{n} f_i(y). \tag{2}$$

Then

$$\max_i \overline{f}(X_i) \leq_{icx} \max_i f_i(X_i). \tag{3}$$

Proof:

To prove the claim, we construct random variable $Y = \max_i f_{\pi(i)}(X_i)$ where π is a randomly chosen permutation $\pi \in \mathcal{P}$. Note that since $X_1, \dots, X_n$ are exchangeable random variables, $Y =_{st} \max_i f_i(X_i)$. Furthermore, Y satisfies the following:

$$\begin{aligned}
& E(Y | \max_i \overline{f}(X_i)) \\
= \;& E_{(X_1,\dots,X_n)|\max_i \overline{f}(X_i)} \left(E(Y|X_1, \dots, X_n, \max_i \overline{f}(X_i)) \right) \\
= \;& E_{(X_1,\dots,X_n)|\max_i \overline{f}(X_i)} \left(E(Y|X_1, \dots, X_n) \right) \\
= \;& E_{(X_1,\dots,X_n)|\max_i \overline{f}(X_i)} \left(E(\max_i f_{\pi(i)}(X_i)|X_1, \dots, X_n) \right) \\
= \;& E_{(X_1,\dots,X_n)|\max_i \overline{f}(X_i)} \left(\frac{1}{n!} \sum_{\pi \in \mathcal{P}} \max_i f_{\pi(i)}(X_i) \right) \\
\geq \;& \frac{1}{n!} E_{(X_1,\dots,X_n)|\max_i \overline{f}(X_i)} \left(\max_i \sum_{\pi \in \mathcal{P}} f_{\pi(i)}(X_i) \right) \\
= \;& \frac{1}{n!} E_{(X_1,\dots,X_n)|\max_i \overline{f}(X_i)} \left(\max_i \left((n-1)! \sum_j f_j(X_i) \right) \right) \\
= \;& E_{(X_1,\dots,X_n)|\max_i \overline{f}(X_i)} \left(\max_i \overline{f}(X_i) \right) \\
= \;& \max_i \overline{f}(X_i).
\end{aligned} \tag{4}$$

The assertion of the proposition is, now, a result of (1).

■

Using similar arguments, the proposition can be extended to a bivariate case:

Proposition 2: Let $X_1, \dots, X_n$ and $T_1, \dots, T_n$ be two independent sets of exchangeable random variables and $f_1, \dots, f_n$ measurable real functions on $\mathbb{R}^2$. Define the function $\overline{f}$ by

$$\overline{f}(y, t) = \frac{1}{n} \sum_{i=1}^{n} f_i(y, t). \tag{5}$$

Then

$$\max_i \overline{f}(X_i, T_i) \leq_{icx} \max_i f_i(X_i, T_i). \tag{6}$$

We end this section with some monotonicity results for queueing systems. We first remind the readers of the recursion formula for waiting time W_l of the lth customer in a G/G/1 queue:

$$W_{l+1} = [W_l + B_l - A_{l+1}]^+ = [W_l + X_l]^+ := \phi(W_l, X_l) \tag{7}$$

where $[y]^+ = \max(0, y)$, A_l is the l^{th} inter-arrival time, B_l is the service time for the lth customer, and $X_l = B_l - A_{l+1}$ denotes the difference between the two quantities. It is known that function $\phi(w, x)$ is an increasing, convex and supermodular function of (w, x) (see [20], page 221). This can be used to establish the following fact for a GI/GI/1 queue (where the inter-arrival and service times are independent).

Fact 5 (Theorem 6.3.2 of [20]) Consider two GI/GI/1 queues both starting from the empty state and with the difference random variables, X' and X'' such that $X' \leq_{icx} X''$, then the waiting times for the lth customers, W'_l and W''_l, follow the same order, i.e.,

$$W'_l \leq_{icx} W''_l,$$

and if stationary waiting times exist, we have the ordering of stationary waiting times:

$$W' \leq_{icx} W''.$$

Now consider a system of n coupled GI/GI/1 queues, where arrivals occur in a synchronized batch. In such a system, the vector of waiting times of the lth batch, $\boldsymbol{W}_l$, follows a multi-dimensional version of (7):

$$\boldsymbol{W}_{l+1} = [\boldsymbol{W}_l + \boldsymbol{B}_l - \boldsymbol{A}_l]^+ = [\boldsymbol{W}_l + \boldsymbol{X}_l]^+ := \boldsymbol{\phi}(\boldsymbol{W}_l, \boldsymbol{X}_l). \tag{8}$$

With similar arguments, the following fact is obtained:

Fact 6: Consider two systems of n coupled GI/GI/1 queues both starting from the empty state, and with batch arrivals to queues, and with the difference random vectors, $\boldsymbol{X}'$ and $\boldsymbol{X}''$ such that $\boldsymbol{X}' \leq_{icx} \boldsymbol{X}''$, then the vector of waiting times for the lth customers, $\boldsymbol{W}'_l$ and $\boldsymbol{W}''_l$, follow the same order, i.e.,

$$\boldsymbol{W}'_l \leq_{icx} \boldsymbol{W}''_l,$$

and if stationary waiting times exist, we have the ordering of stationary waiting times:

$$\boldsymbol{W}' \leq_{icx} \boldsymbol{W}''.$$

III. Problem Formulation

A. A Simple Queueing Model

In order to analyze the delay performance of various sharing schemes we use the simple queueing model shown in Fig. 1, consisting of n queues and $2n$ servers.

We assume that n stream of jobs arrive at n primary servers according to n independent processes. These primary servers are responsible for forwarding the jobs to the secondary queues. They are allowed to divide and distribute the jobs across the servers. Specifically, primary servers forward the jobs to secondary servers according to a job *distribution* policy. Jobs are queued at a secondary queue when the corresponding secondary server is busy. The forwarding, division, and distribution of jobs are done instantaneously, i.e., service rates of the primary servers are infinite. In addition, we assume that these actions are performed without any knowledge of the state of the system. Secondary servers are assumed to be statistically identical, and jobs leave the system after completing service at secondary queues. We consider a class of (time invariant) job distribution policies as follows:

(A1) Jobs arrive at n primary servers according to n independent Poisson processes each with rate λ.

(A2) Jobs are distributed according to a stationary *distribution policy* across secondary servers.

(A3) Each distribution policy is defined according to a *distribution matrix* $A = \begin{bmatrix} \boldsymbol{\alpha}^1 \\ \vdots \\ \boldsymbol{\alpha}^n \end{bmatrix}$, which is stochastic and whose i^{th} row, $\boldsymbol{\alpha}^i$, indicates how primary server i distributes jobs across secondary servers.

(A4) Given the distribution matrix $A = \begin{bmatrix} \boldsymbol{\alpha}^1 \\ \vdots \\ \boldsymbol{\alpha}^n \end{bmatrix}$ a job arriving at the primary server i is distributed across secondary queues such that the service time of pieces form a random vector $(\alpha^i_1 \tau_1, \alpha^i_2 \tau_2, \ldots, \alpha^i_n \tau_n)$.

(A5) Random variables $\tau_1, \tau_2, \ldots, \tau_n$ are associated and exchangeable with marginal distribution G and finite first three moments and $\lambda E(\tau) := \lambda \int \tau dG < 1$.

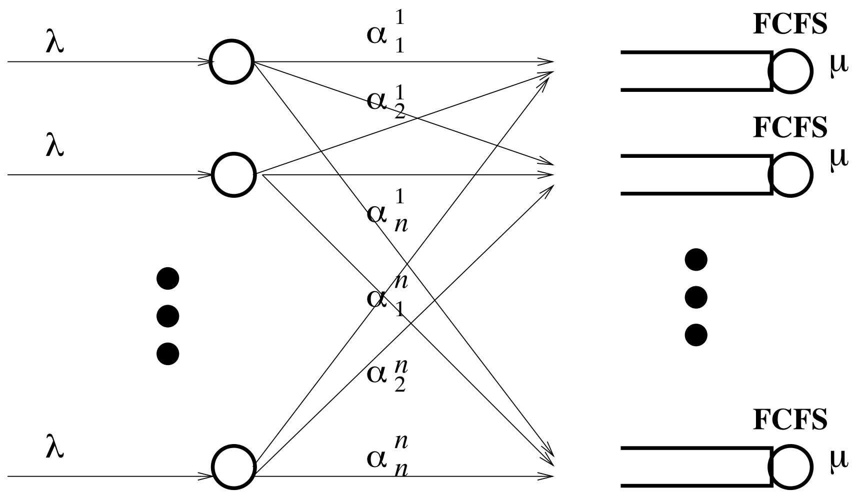

Fig. 1. A simple queueing model for resource sharing among homogeneous users

(A6) Forwarding and division of jobs are done with zero delay and in a time invariant manner (with no knowledge of secondary queue backlogs).
(A7) Service times and inter-arrival times are independent and identically distributed across time.

We are interested in finding the "optimal" A for a given λ and G, where the performance measure of interest is the expected *session delay*, also known as *application layer delay*.

Definition 6: The *session delay* of a job is the delay the job sees from the moment it arrives at a primary server until its last piece finishes service at a secondary server.

It is easy to see that the delay for a given job in the stream i (in steady state) is given by

$$D_i(A) = \max_{j:\alpha_j^i \neq 0} \left\{ \alpha_j^i \tau_j + W_j(A) \right\} \tag{9}$$

where $W_j(A)$, $j = 1, \dots, n$, is the stationary waiting time at the secondary queue j, under policy A.

We are also interested in minimizing $E(D_i(A))$ for a given i over choices of distribution matrix. Before formally introducing the problem statement, we need the following definition:

Definition 7: A policy A is called *symmetric* if it is doubly stochastic and divides the jobs in such a way that the division vectors α^i for primary queues $i, i = 1, \dots, n$ are permutations of each other. We denote the class of such policies by $\mathcal{A}_s$.

Note that restricting attention to $\mathcal{A}_s$ ensures that the load of each server under sharing remains the same for all choices of $A \in \mathcal{A}$ and in particular, identical to the load distribution under the exclusive allocation of streams to secondary servers. This is a reasonable assumption as it ensures that, in case of server ownership, cooperation does not cause a load increase on any one user's resource.

For the reasons mentioned earlier, we consider distinct (but related) problems (P1)-(P3). Problems (P1)-(P2) assume zero sharing cost, while (P3) investigates the issue of sharing cost.

P1 Show that for any job arriving at the system, among all symmetric policies, equal sharing matrix $A^* = [1/n]_{n\times m}$ minimizes (in an increasing convex stochastic order sense) all streams' session delays; i.e., for all jobs,

$$D_i(A^*) \leq_{icx} D_i(A), \qquad \text{for } \forall i, \forall A \in \mathcal{A}_s.$$

P2 Show that in a non-cooperative game in which each user chooses its distribution $\boldsymbol{\alpha}^i$ autonomously, the optimal distribution policy A^* is also a Nash equilibrium.

P3 Show that when sharing incurs a cost, the asymptotic behavior of the optimal policy strongly depends on the model of overhead.

Since $\mathcal{A}_s$ is nothing but the set of all matrices A such that A is a doubly stochastic matrix whose rows are permutations of each other, we simplify notation by replacing A with its first row and a collection of n corresponding permutations of that first row. For instance, we can write $D_{i,P}(\boldsymbol{\alpha}) := D_i(A(\boldsymbol{\alpha}, P))$, where $A(\boldsymbol{\alpha}, P)$ is a doubly stochastic matrix whose l^{th} row is the $\pi^l \in P \subset \mathcal{P}$ permutations of $\boldsymbol{\alpha}$, i.e., $A(\boldsymbol{\alpha}, P) := \begin{bmatrix} \pi^1(\boldsymbol{\alpha})=\boldsymbol{\alpha} \\ \vdots \\ \pi^n(\boldsymbol{\alpha}) \end{bmatrix}$. Similarly, we write $W_{j,P}(\boldsymbol{\alpha}) := W_j(A(\boldsymbol{\alpha}, P))$.

B. Related Work

Our first problem (P1) is similar to the one studied in [24], [9], and [16]. The one major difference between our model and that studied in these papers is the availability of queue backlog information at the primary servers. In these papers, server backlog information is available to a centralized controller, while we consider the problem where the primary server's decision is independent of such information. We note that the seemingly contradictory result with those presented in [16] is not surprising when one considers the value of state information. The sequential policy in [16] is different from the policy that forwards the jobs exclusively to a particular server. Indeed, the sequential policy in [16] can identify and join the secondary queue with minimum backlog, while such information regarding backlogs is not available to the servers in our setting.

Our problem is also related to a classical task assignment (also known as routing) problem in which arriving jobs are to be assigned to one out of many available servers. (For a complete discussion on task assignment problems with various assumptions, see [11] and the references therein.) However, our problem differs from the classical task assignment in two aspects. The first is that we allow for jobs to be broken into smaller pieces, potentially with some added redundancy/overhead cost. The second difference is more philosophical. In this paper, we restrict our attention to a subset of policies which result in *symmetric sharing*. Unlike the traditional task assignment problems where all servers are owned and managed by an entity whose interest is distinct from those of the arriving streams, we are interested in cases where there is an association between a particular (secondary) server and a particular stream of jobs. In this case, we show that it is far more reasonable to restrict the notion of cooperation to symmetric sharing, protecting any user from non-cooperative behavior. For instance, consider the problem of content distribution in P2P networks: it is intuitive that a symmetric cooperation among hosts is far more reasonable than an arbitrary and asymmetric cooperation which can leave some users potentially vulnerable to selfish behavior. We concretize this matter, when addressing P2, via the notion of Nash equilibrium.

After identifying policy of A^* as the optimal sharing policy in $\mathcal{A}_s$, under zero sharing cost assumption, we establish a simple result for the case of non-zero sharing costs. Specifically, we compare policy A^* with a non-sharing policy in which each stream of jobs is allocated exclusively to one particular server. Our result establishes the continuity of delay with respect to sharing parameters to arrive at an existence result: we show that for any admissible load, there always exists a level of acceptable sharing cost below which A^* outperforms exclusive allocation of servers. This result is a generalization of the performance analysis of the classical fork/join queueing problems ([3], [4], [5], [14], [21], and [22]) with an addition of sharing cost to the model.

We would also like to point out that this paper builds upon our prior work [13], where we assumed independent service times across secondary servers while ignoring the issue of sharing cost.

IV. Analysis of Problem P1

In this section, we restrict our attention to distribution matrices $A \in \mathcal{A}_s$. Theorem 1 below provides the first result of our paper.

Theorem 1: Assume G and λ are such that $\lambda E(\tau_i) < 1$. For any given job of stream i, any permutation collection $P \in \mathcal{P}$ and distribution vector $\boldsymbol{\alpha}$ such that $A(\boldsymbol{\alpha}, P) \in \mathcal{A}_s$, we have:

$$D_i([1/n]) \leq_{icx} D_{i,P}(\boldsymbol{\alpha}).$$

Proof: Without loss of generality, we consider a job arriving at the primary server 1. Also, we use $D_i(1/n)$, $D_i([1/n])$, and $D_{i,P}(1/n)$ interchangeably.

Remember that given vector $\boldsymbol{\alpha}$ and the appropriate permutation set P,

$$D_{1,P}(\boldsymbol{\alpha}) = \max_{j:\boldsymbol{\alpha}_j \neq 0} \left\{ \alpha_j \tau_j + W_{j,P}(\boldsymbol{\alpha}) \right\},$$

and

$$D_1(1/n)) = \max_j \left\{ \frac{\tau_j}{n} + W_j(1/n) \right\}.$$

It is our goal to prove

$$D_1(1/n)) = \max_j \left\{ \frac{\tau_j}{n} + W_j(1/n) \right\} \leq_{icx} \max_{j:\boldsymbol{\alpha}_j \neq 0} \left\{ \alpha_j \tau_j + W_{j,P}(\boldsymbol{\alpha}) \right\} = D_{1,P}(\boldsymbol{\alpha})).$$

Note that here, due to the symmetry among servers, as well as restriction to symmetric policies, we can restrict our attention to $\boldsymbol{\alpha}$ such that $\sum_j \alpha_j = 1$, $\alpha_j \geq \alpha_{j+1}$.

To prove this theorem we use four steps.

Step 1. (Proposition 3) For $m = \#(\{\alpha_j : \alpha_j \neq 0\}) = \max\{j : \alpha_j > 0\}$ and a set P of appropriate permutations, we have

$$(W_{1,P}(\overbrace{\frac{1}{m} \cdots \frac{1}{m}}^{m} \; 0 \cdots 0) \; \ldots \; W_{m,P}(\overbrace{\frac{1}{m} \cdots \frac{1}{m}}^{m} \; 0 \cdots 0)) \leq_{st} (W_{1,P}(\boldsymbol{\alpha}) \; \ldots \; W_{m,P}(\boldsymbol{\alpha})).$$

<u>Notation:</u> From now on, in place of $W_{i,P}((\overbrace{1/m \cdots 1/m}^{m} \; 0 \cdots 0)$, we abuse the simpler notation $W_{i,P}(1/m)$.

Step 2. (Lemma 3) $\max_{1 \leq j \leq m} \left\{ \frac{\tau_j}{m} + W_{j,P}(1/m) \right\} \leq_{icx} \max_{j:\boldsymbol{\alpha}_j \neq 0} \left\{ \alpha_j \tau_j + W_{j,P}(\boldsymbol{\alpha}) \right\}$

Step 3. (Lemma 4) $\max_{1 \leq j \leq n} \left\{ \frac{\tau_j}{n} + \frac{m}{n} W_{j,P}(1/m) \right\} \leq_{icx} \max_{1 \leq j \leq m} \left\{ \frac{\tau_j}{m} + W_{j,P}(1/m) \right\}$

Step 4. (Lemma 5) For any $1 \leq k \leq n$, we have $kW_{i,P}(1/k) =_{st} nW_{i,P}(1/n)$. Furthermore,

$$\max_{1 \leq j \leq n} \left\{ \frac{\tau_j}{n} + W_{j,P}(1/n) \right\} \leq_{icx} \max_{1 \leq j \leq n} \left\{ \frac{\tau_j}{n} + \frac{m}{n} W_{j,P}(1/m) \right\}$$

Following the above steps, we have the assertion of the theorem.

■

The following corollary is a direct consequence:

Corollary 1: For any job arriving at the system, $A^* = [1/n]_{n \times n}$ minimizes delay in all moments, i.e., for $\forall r \in \mathbb{N}$

$$E([D_i(1/n)]^r) \leq E([D_{i,P}(\boldsymbol{\alpha})]^r).$$

In what comes next, we provide proof for Proposition 3 as well as Lemmas 3-5. In order to prove Proposition 3, we provide the following simple lemmas whose proofs are omitted for brevity (see [13] for details).

Lemma 1: Consider l^{th} job arriving at the system, under distribution policy $\boldsymbol{\alpha}$ and permutation set P. This will generate jobs at the secondary servers whose service times form a random vector $(\tau^1_{\boldsymbol{\alpha},P}(l), \ldots, \tau^n_{\boldsymbol{\alpha},P}(l))$, where $\tau^j_{\boldsymbol{\alpha},P}(l)$ is service time of the piece arriving at secondary queue j. We have

$$(\tau^1_{\boldsymbol{\alpha},P}(l) \; \ldots \; \tau^j_{\boldsymbol{\alpha},P}(l) \; \ldots \; \tau^n_{\boldsymbol{\alpha},P}(l)) \geq_{cx} (\tau^1_{1/n}(l) \; \ldots \; \tau^j_{1/n}(l) \; \ldots \; \tau^n_{1/n}(l)).$$

Proof of this lemma is directly based on (A1), the definition of convex ordering of stochastic vectors (see [13] for details), and by noting that random vector $(\tau^1_{\boldsymbol{\alpha},P}(l), \ldots, \tau^n_{\boldsymbol{\alpha},P}(l))$ is equal to $(\alpha_{\pi^i(1)}\tau_1, \ldots, \alpha_{\pi^i(n)}\tau_n)$ with equal probability $1/n$, where $\pi^i, i = 1, 2, \ldots, n$ are permutations $\pi^i \in \mathcal{P}$ describing how A and $\boldsymbol{\alpha}$ are related.

Lemma 2: For $m = \#(\{\alpha_j : \alpha_j \neq 0\}) = \max\{j : \alpha_j > 0\}$

$$(\tau^1_{\boldsymbol{\alpha},P}(l) \ \dots \ \tau^j_{\boldsymbol{\alpha},P}(l) \ \dots \ \tau^n_{\boldsymbol{\alpha},P}(l)) \geq_{cx} (\tau^1_{1/m,P}(l) \ \dots \ \tau^j_{1/m,P}(l) \ \dots \ \tau^n_{1/m,P}(l)).$$

Note that from the symmetric sharing assumption and the construction of $\mathcal{A}_s$, random service time $\tau^j_{\boldsymbol{\alpha},P}(l)$ is always non-zero, i.e. there exists at least one primary queue that forwards its packets to secondary queue j.

Now we are ready to (restate and) prove Proposition 3 and Lemmas 3-5.

Proposition 3: For $m = \#(\{\alpha_j : \alpha_j \neq 0\}) = \max\{j : \alpha_j > 0\}$, we have

$$(W_{1,P}(1/m) \ \dots \ W_{m,P}(1/m)) \leq_{icx} (W_{1,P}(\boldsymbol{\alpha}) \ \dots \ W_{m,P}(\boldsymbol{\alpha})).$$

Proof: The proof of this proposition is based on Lemmas 1 and 2 and Fact 6. ■

Now we use the above proposition to establish the following:

Lemma 3: For $m = \#(\{\alpha_j : \alpha_j \neq 0\}) = \max\{j : \alpha_j > 0\}$,

$$\max_{1\leq j\leq m} \left\{\frac{\tau_j}{m} + W_{j,P}(1/m)\right\} \leq_{icx} \max_{j:\boldsymbol{\alpha}_j\neq 0} \{\alpha_j\tau_j + W_{j,P}(\boldsymbol{\alpha})\} \tag{10}$$

Proof: The proof of this lemma is based on the following two inequalities:

$$\max_{j:\boldsymbol{\alpha}_j\neq 0} \{\alpha_j\tau_j + W_{j,P}(1/m)\} \quad \leq_{icx} \quad \max_{j:\boldsymbol{\alpha}_j\neq 0} \{\alpha_j\tau_j + W_{j,P}(\boldsymbol{\alpha})\}, \tag{11}$$

$$\max_{1\leq j\leq m} \left\{\frac{\tau_j}{m} + W_{j,P}(1/m)\right\} \quad \leq_{icx} \quad \max_{j:\boldsymbol{\alpha}_j\neq 0} \{\alpha_j\tau_j + W_{j,P}(1/m)\}. \tag{12}$$

The first inequality is a direct result of Lemma 2 and Fact 3 when $g(u, v) = \max_i(u_i + v_i)$, $X = W_P(1/n)$, and $Y = W_P(\boldsymbol{\alpha})$, $Z = (\alpha_1\tau_1, \dots, \alpha_m\tau_m)$. On the other hand, the second inequality is based on Proposition 2 as we use the exchangeability of $W_{j,P}(1/m)$ and τ_j while putting $f_j(x, t) := \alpha_j t + x$ (hence, $\overline{f}(x) = t/m + x$). Using (1), we arrive at the assertion of the lemma.

■

Similarly, we have the following lemma:

Lemma 4: For all $1 \leq m \leq n$,

$$\max_{1\leq j\leq n} \left\{\frac{\tau_j}{n} + \frac{m}{n}W_{j,P}(1/m)\right\} \leq_{icx} \max_{1\leq j\leq m} \left\{\frac{\tau_j}{m} + W_{j,P}(1/m)\right\}$$

Proof: The proof of this lemma is based on Proposition 2 and exchangeability of $W_{j,P}(1/m)$ and τ_j. Let for all $j = 1, 2, \dots, n$,

$$f_j(x, t) := \begin{cases} t/m + x & j \leq m \\ 0 & j > m. \end{cases}$$

It is clear that $\overline{f}(x, t) = \frac{t}{n} + \frac{m}{n}x$ and with Proposition 2 we have the assertion of the lemma. ■

Lemma 5: For any $1 \leq k \leq n$, we have $kW_i(1/k) =_{st} nW_i(1/n)$. Furthermore,

$$\max_{1\leq j\leq n} \left\{\frac{\tau_j}{n} + W_{j,P}(1/n)\right\} \leq_{icx} \max_{1\leq j\leq n} \left\{\frac{\tau_j}{n} + \frac{m}{n}W_{j,P}(1/m)\right\}.$$

Proof: It is well known (e.g., see Theorem 5.7, page 237, in [2]) that in an M/GI/1 queue, the stationary waiting time W, given that it exists, is given by $W =_{st} \sum_{l=1}^N X_l$, where N is a geometrically distributed random variable with parameter $\rho = \lambda m_b$ (m_b is the mean service time and λ is the arrival rate) while $X_1, X_2, \dots$ are i.i.d random variables with distribution $F_e(t) = \frac{1}{m_b}\int_0^t (1 - F_b(x))dx$. From this, we have the waiting time $W_1(1/k)$ (and $W_i(1/k)$), as

$$W_1(1/k) =_{st} \sum_{l=1}^{N} X_l(k). \tag{13}$$

where N is a geometrically distributed random variable with parameter $\rho = k\lambda E(\tau/k) = \lambda E(\tau)$, and $X_l(k)$ is distributed according to

$$F_k(t) = \frac{1}{E(\tau/k)} \int_0^t (1 - F_{\tau/k}(x))dx = \frac{k}{E(\tau)} \int_0^t (1 - F_\tau(kx))dx = F_1(kt). \tag{14}$$

In other words, $kW_1(1/k) =_{st} k \sum_{l=1}^{N} X_l(k) = \sum_{l=1}^{N} kX_l(k) = \sum_{l=1}^{N} Y_l$, where the distribution function of Y_l is $P(Y_l \leq t) = P(X_l(k) \leq t/k) = F_1(t)$. This implies that the distribution of $kW_1(1/k)$ is independent of k, hence the first part of the lemma.

Now using the fact that vectors of $(mW_{1,P}(1/m), \ldots, mW_{n,P}(1/m)$ have marginal distributions independent of m, and the fact that the correlation among waiting times in queues grows with m (when $m = 1$, the queues are independent), we arrive at the following

$$(nW_{1,P}(1/n), \ldots, nW_{n,P}(1/n) \leq_c (kW_{1,P}(1/k), \ldots, kW_{n,P}(1/k)), \tag{15}$$

where $\leq_c$ is the concordance order (see [20], page 108). Intuitively, the concordance order can be thought of as a measure of dependence among elements of two random vectors with similar marginals. Equation (15) implies that the joint CDF of random vectors $(kW_{1,P}(1/k), \ldots, kW_{n,P}(1/k))$ is monotonically decreasing in k, $k = 1, \ldots, n$. Now, using theorems 3.3.17 and 3.3.18 of [20], we have the assertion of the lemma. ■

So far we have demonstrated that among symmetric sharing policies, the policy which distributes the jobs equally among all servers is optimal in an increasing convex stochastic order sense. The next section tackles user incentives to cooperate.

V. Analysis of Problem P2

In this section, we show that when sharing does not incur a cost, all equal sharing also constitutes a Nash equilibrium. In other words, given that all users, but one, follow an all-equal splitting rule (and in the absence of queue backlog information), the average delay of the remaining user can only increase if she chooses to deviate from the all-equal splitting rule. Before we start, we need the following notations:

$$D_j(\boldsymbol{\alpha}^i, \boldsymbol{\alpha}^{*-i}) := D_j(\begin{bmatrix} 1/n \cdots 1/n \\ \vdots \\ \boldsymbol{\alpha}^i \\ \vdots \\ 1/n \cdots 1/n \end{bmatrix}) \qquad W_j(\boldsymbol{\alpha}^i, \boldsymbol{\alpha}^{*-i}) := W_j(\begin{bmatrix} 1/n \cdots 1/n \\ \vdots \\ \boldsymbol{\alpha}^i \\ \vdots \\ 1/n \cdots 1/n \end{bmatrix}),$$

where $D_j(\boldsymbol{\alpha}^i, \boldsymbol{\alpha}^{*-i})$ denotes the random (stationary) delay an arbitrary job arriving at the j^{th} primary queue experiences, when user i follows the distribution policy $\boldsymbol{\alpha}^i$ ($\boldsymbol{\alpha}^i.\mathbf{1} = 1$) while all other users follow the all equal distribution policy, $\boldsymbol{\alpha}^{*-i} = [1/n]_{(n-1)\times n}$. Similarly, $W_j(\boldsymbol{\alpha}^i, \boldsymbol{\alpha}^{*-i})$ denotes the random (stationary) waiting time in secondary queue j.

It is not hard to see that

$$D_i(\boldsymbol{\alpha}^i, \boldsymbol{\alpha}^{*-i}) \quad = \quad \max_j \left\{\alpha_j^i \tau_j + W_j(\boldsymbol{\alpha}^i, \boldsymbol{\alpha}^{*-i})\right\} \tag{16}$$

The main result of this section is articulated in the following theorem:

Theorem 2: Consider the primary server i. All equal distribution of the jobs among queues constitutes a Nash equilibrium in an average delay sense, i.e.,

$$E(D_i(\boldsymbol{\alpha}^{*i}, \boldsymbol{\alpha}^{*-i})) \leq E(D_i(\boldsymbol{\alpha}^i, \boldsymbol{\alpha}^{*-i})). \tag{17}$$

Proof: Without loss of generality, let $i = 1$. Let $\boldsymbol{\alpha}$ denote the distribution strategy of primary server 1.

Note that, unlike the case in Theorem 1, under distribution matrix $(\boldsymbol{\alpha}, \boldsymbol{\alpha}^{*-1})$, one cannot simply order (in a usual stochastic or convex sense) the random vectors of service times across secondary queues in order to appeal to Fact 4. To see this, notice that all users but the first one follow a symmetric distribution which guarantees that user 1's unequal distribution strategy increases (in a stochastic sense) the service time in one secondary queue while decreasing it in another. This, together with the fact that usual stochastic order (convex order) is closed with respect to marginalization, indicates that the vector of service times under $(\boldsymbol{\alpha}, \boldsymbol{\alpha}^{*-1})$ cannot be stochastically ordered relative to the vector of service times under A^*. So instead of ordering the vectors of service times, we use Fact 1 to prove this theorem.

In what comes next, we use Fact 1 to show that $E(D_1^s(\boldsymbol{\alpha}, \boldsymbol{\alpha}^{*-1})) = E(\max_j(\alpha_j \tau_{1,j}^s + W_j^s(\boldsymbol{\alpha}, \boldsymbol{\alpha}^{*-i})))$ is symmetric and convex in $\boldsymbol{\alpha}$. Since $\phi(w_1, w_2, \ldots, w_n) = \max_j(w_j)$ is convex, increasing, and symmetric on $\mathbb{R}^n$, we focus on the delay of the j^{th} piece of a job arriving originally at primary queue 1 and subsequently distributed and rerouted to secondary queue j:

$$w_j = \alpha_j \tau_j^s + W_j^s(\boldsymbol{\alpha}, \boldsymbol{\alpha}^{*-1}),$$

where $\alpha_j \tau_{1,j}^s$ is the service time (with zero duration if $\alpha_j^1 = 0$) of a job piece, while $W_j^s(\boldsymbol{\alpha}, \boldsymbol{\alpha}^{*-1})$ is the waiting time for this job piece in secondary queue j. Note that here the superscript s imply that the job piece of interest belongs to the s^{th} job arriving to the overall system. Furthermore, note that the service time for all job pieces originated and distributed from primary queue $i, i = 2, 3, \ldots, n$, and rerouting to secondary queue j are statistically identical to $\frac{1}{n}\tau_j$.

Next, we show that for $\forall s$, $W_j^s(\boldsymbol{\alpha}, \boldsymbol{\alpha}^{*-1})$ (hence, $\alpha_j \tau_j^s + W_j^s(\boldsymbol{\alpha}, \boldsymbol{\alpha}^{*-1})$) is a function of α_j as well as a set of random variables $\{V_j^\sigma\}_{\sigma=1}^{2s}$ such that:
(i) W_j^s is convex in α_j for any given and fixed realization of $\{V_j^\sigma\}_{\sigma=0}^{2s}$, and
(ii) For $\forall \sigma$, $(V_1^\sigma, \ldots, V_j^\sigma, \ldots, V_n^\sigma)$ is an exchangeable random vector.

To show (i), we note that $W_j^\sigma(\boldsymbol{\alpha}, \boldsymbol{\alpha}^{*-1})$ is the solution to the following Lindley equation:

$$W_j^{\sigma+1} = [W_j^\sigma + B_j^\sigma - A_j^{\sigma+1}]^+ = [W_j^\sigma + X_j^\sigma]^+ := \phi(W_j^\sigma, X_j^\sigma), \quad (18)$$

where $[y]^+ = \max(0, y)$, A_j^σ is the σ^{th} job inter-arrival time into secondary queue j, and B_j^σ is the service time for the σ^{th} customer in secondary queue j, i.e.,

$$B_j^\sigma = \begin{cases} \frac{1}{n}\tau_j^\sigma & \text{if } \sigma^{th} \text{ job arrived at primary queue } i \neq 1 \\ \alpha_j \tau_j^\sigma & \text{if } \sigma^{th} \text{ job arrived at primary queue } 1 \end{cases}. \quad (19)$$

Let us clarify that B_j^σ is a function of α_j, while A_j^σ, is not. More precisely, we note that when $\alpha_j = 0$, upon a job arrival to primary queue 1, still a job piece (with zero service time) arrives at the j^{th}secondary queue[1]. In other words, upon the σ^{th} arrival into the system, there are simultaneous arrivals (of possibly zero-sized jobs) into all secondary queues.

Now, put $V_j^\sigma = \begin{cases} A_j^k & \text{if } \sigma = 2k \\ \tau_j^k & \text{if } \sigma = 2k+1 \end{cases}$. The convexity of W_j^s in α_j for any given realization of $\{V_j^\sigma\}_{\sigma=1}^{2s}$ follows directly from (18) and the convexity, increasing, and supermodularity properties of ϕ (see [20], page 221).

What remains is to verify that for $\forall \sigma$, $(V_1^\sigma, V_2^\sigma, \ldots, V_n^\sigma)$ is an exchangeable random vector. This is trivially true by construction. In other words, because arrivals into the secondary queues are simultaneous and with zero delay (A6), i.e. for all j, k, σ, $A_j^\sigma = A_k^\sigma = A^\sigma$, where A^σ is the σ^{th} inter-arrival time into the overall system; while similarly, for any given σ, $(\tau_1^\sigma, \ldots, \tau_j^\sigma, \ldots, \tau_n^\sigma)$ are assumed to be exchangeable (A5).

Applying Fact 1 sequentially, for independent (A7) exchangeable random vectors $\mathbf{V}^\sigma$, $\sigma = 0, \ldots, 2s$, we have that

$$E\left(D_1^s(\boldsymbol{\alpha}, \boldsymbol{\alpha}^{*-1})\right) = E\left(\max_j \left\{\alpha_j \tau_j^s + W_j^s(\boldsymbol{\alpha}, \boldsymbol{\alpha}^{*-1})\right\}\right) = E\left(\max_j \left\{w(\{V_j^\sigma\}_{\sigma=0}^{2s}, \alpha_j)\right\}\right) = \psi(\boldsymbol{\alpha}) \quad (20)$$

[1]Lindley's equation remains valid for service distributions with atoms at zero.

is a symmetric and convex function in $\boldsymbol{\alpha}$. Given the constraint $\boldsymbol{\alpha}\mathbf{1} = 1$, the symmetric convexity of ψ implies that it is minimized at $\boldsymbol{\alpha}^* = (1/n, \ldots, 1/n)$, hence the assertion of the theorem. ■

Now that we have established the optimality of sharing policy, both in the context of symmetric sharing policies as well as in a Nash equilibrium sense, we move to investigate the impact of overhead increase or added redundancy due to (costs associated with) sharing.

VI. Analysis of Problem P3

In this section, we intend to account for the cost associated with sharing. To do so, we modify our proposed model from Section III as follows:

(A4') Given distribution policy A and sharing cost B, we define the service time of pieces of a job distributed by primary server i as a random vector of the form $(t_1^i(A,B) \ldots, t_j^i(A,B) \ldots t_n^i(A,B))$.

In particular, we consider the following two models of sharing cost:

Model (M1) In this model, we assume that the cost of sharing is dominated by overhead increase, i.e., $t_j^i(A, B_\delta) := \alpha_j^i(\tau_j - \delta) + \delta$, $\delta \geq 0$. Note that this model is appropriate to model the network layer overhead increase, such as addition of IP headers, etc., since sharing cost increases linearly with the number of pieces into which a job is broken.

Model (M2) In this model, we assume that the cost of sharing is dominated by redundancy increase prior to distribution of job among servers, i.e., $t_j^i(A, B_\beta) := \beta\alpha_j^i\tau_j$, $\beta \geq 1$. Note that this model is appropriate to model coding cost (associated with multi-description coding and multi-path routing as discussed in [8]).

Note: Under (M1), we only need to consider the number of servers among which the job is divided, since the optimality of $(1/m, \ldots, 1/m)$ among all sharing of m servers is a corollary to Theorem 1. Under (M2), on the other hand, all strict sharing policies are inferior to $A^* = [1/n]$. This is because under (M2) coding is done prior to distribution of the job among secondary servers, hence, sharing policies (post coding) all have equal overhead and load increase.

We are interested in comparing $E(D_i(1/n))$ with $E(D_i^{ex})$, where the superscript of ex refers to an exclusive allocation which forwards all jobs of stream i to server i. The main results of this section are given by the following three theorems:

Theorem 3: Given a load $\rho < 1$ for the non-sharing system, there exists a small enough sharing cost (β or δ), under which the all-equal sharing policy outperforms the exclusive allocation policy.

Proof: Let us consider Model (M2). In the first step of the proof, we compare the average length of a secondary queue under both policies, using the Pollaczek-Khinchin formula to compute $E(W(1/n))$ and $E(W^{ex})$ (note that each secondary queue i is an M/G/1 queue):

$$E(W^{ex}) = \frac{\lambda E(\tau^2)}{2(1-\rho)}, \qquad \text{and} \qquad E(W_j(\frac{1}{n})) = \frac{\lambda\beta^2 E(\tau^2)}{2n(1-\beta\rho).} \tag{21}$$

Now we use this to bound the delay associated with each job from below and above:

$$\beta E(\tau)/n + E(W_j(\frac{1}{n})) \;\leq\; E(D_1(\tfrac{1}{n})) \;\leq\; E(\max_{1\leq j\leq n}\beta\tau_j)/n + E(\max_{1\leq j\leq n}(W_j(\frac{1}{n}))) \tag{22}$$

Now we use (21) and the simple bound of $E(\max_{1\leq j\leq n} X_j) < nE(X_j)$, when $n \geq 2$ and $E(X_j) \neq 0$, to get

$$\beta E(\tau)/n + \frac{\lambda\beta^2 E(\tau^2)}{2n(1-\beta\rho)} \;\leq\; E(D_1(\tfrac{1}{n})) \;<\; \beta E(\max_{1\leq j\leq n}\tau_j)/n + \frac{\lambda\beta^2 E(\tau^2)}{2(1-\beta\rho)} \tag{23}$$

Thereby showing that for any given load $\rho < 1$, $E(D_1(1/n))$ is always bounded by a continuous function at some small neighborhood of $\beta = 1$. Furthermore, we have shown that when $\beta = 1$, $E(D_i(\frac{1}{n})) < E(D_i^{ex})$. From this we have the assertion of the theorem for Model (M2).

The proof for Model (M1) is similar and is omitted in the interest of brevity. ■

So far we have demonstrated that there always exists a small enough overhead/redundancy such that sharing outperforms non-sharing policies. For a given sharing cost, though, it is easy to construct an example with a high traffic load in which the increase in network load, caused by overhead/redundancy, makes sharing inferior to the exclusive allocation of jobs to servers. In this aspect, not surprisingly, models (M1) and (M2) exhibit different asymptotic behavior. This is mainly due to the fact that in Model (M1) the total network load increases with the number of server/users, while under Model (M2) the total network load does not change, while variance of service times remains bounded. The following propositions complete our results in this section, by concretely quantifying the difference in the two models:

Proposition 4: For a given load $\rho = \lambda/\mu$ and given overhead δ (M1), there exists N such that for $\forall n > N$, policy $A = [1/n]$ results in an infinite delay; in particular $E(D_i(1/n)) > E(D_i^{ex})$ for $\forall n \geq N$.

Proof: We first derive upper and lower bounds (similar to those in eqn. (22)) for the session delay of a given job under Model (M1). Again, we use the Pollaczek-Khinchin formula to bound the delay of each M/G/1 secondary queue i:

$$E(W^{ex}) = \frac{\lambda E(\tau^2)}{2(1-\rho)}, \qquad \text{while} \qquad E(W_j(1/n)) = \frac{\lambda(E(\tau^2) + 2\delta n(n-1)E(\tau) + \delta^2(n-1)^2}{2n(1-\lambda(E(\tau)+(n-1)\delta))}. \quad (24)$$

Now it is easy to see that, for $N > \frac{1-\rho}{\delta\lambda} + 1$, the added redundancy will cause $\rho' = \lambda' E(\tau') > 1$ resulting in infinite average delay in secondary queues. With this, we have the assertion of the proposition. ■

Proposition 5: For given load $\rho = \lambda/\mu$ and redundancy $\beta > 1$ (Model (M2)) such that $\beta\rho < 1$, there exists N such that for $\forall n \geq N$, $A^* = [1/n]$ results in strictly better average delay, i.e., $E(D_i(1/n)) < E(D_i^{ex})$, $\forall n \geq N$.

Proof: We use Fact 2 and a distribution free upper bound (see Section 4.2. of [10]):

$$E(\max_j X_j) \leq E(\max_j \hat{X}_j) \leq E(X) + \frac{n-1}{\sqrt{2n-1}} \text{Var}^{\frac{1}{2}}(X). \quad (25)$$

where $(X_1, \ldots, X_n)$ is a vector of associated random variables and $(\hat{X}_1, \ldots, \hat{X}_n)$ a vector of iid random variables with the same marginal distribution.

Next we use this upper bound together with (22). We first calculate the second moment of the waiting time in secondary queues (see Theorem 5.7 of [2], page 237):

$$\text{Var}(W_j(1/n)) = \frac{\lambda\beta^3 E(\tau^3)}{3n^2(1-\beta\rho)} + \frac{\lambda^2\beta^4 E^2(\tau^2)}{2n^2(1-\beta\rho)^2}. \quad (26)$$

Now replacing the order statistic upper bounds in (22), we have

$$E(D_1(1/n)) \leq \frac{\beta E(\tau)}{n} + \frac{\beta(n-1)}{n\sqrt{2n-1}} \text{Var}^{\frac{1}{2}}(\tau) + \frac{\lambda\beta^2 E(\tau^2)}{2n(1-\beta\rho)} + \frac{n-1}{n\sqrt{2n-1}} \left(\frac{\lambda\beta^3 E(\tau^3)}{3(1-\beta\rho)} + \frac{\lambda^2\beta^4 E^2(\tau^2)}{2(1-\beta\rho)^2} \right)^{\frac{1}{2}}.$$

Now given $\rho\beta < 1$ and noticing the monotonic decreasing property of the right hand side (with respect to n), we have the assertion of the proposition. ■

VII. Future Work

We have a long way to go before fully characterizing the delay improvements due to resource sharing in a network. In the current paper, we have shown that even in very simple scenarios with parallel servers there exists a sharing cost under which sharing servers in a simple-to-implement way is beneficial and there are conditions under which it is not. This result complements and strengthens known results for networks with bottle-neck links as studied in [6] or, for schemes which increase throughput [15]. Furthermore, we have established a Nash equilibrium property for the symmetric sharing strategy in the case of a homogenous server structure.

One of the weaknesses of the current model is the restriction to homogeneous system assumption in which all servers are identical and all arrival streams have similar characteristics. This assumption is far from reality in all applications of interest. The optimal splitting rule, in the absence of queue backlog information, in such practice will potentially depend on the load as well as the degree of asymmetry among servers and arrival statistics. Following insights from the result of this paper, we conjecture that it is optimal (in an average delay sense) for each job to be distributed in such a way that the random vector associated with various pieces across secondary queues is minimized (in some multivariate stochastic sense). That is, in an asymmetric setting, the result obtained above can be extended to identify the policy which minimizes the average delay. It is important to emphasize a significant difference between the optimality result in the symmetric and asymmetric case: In the symmetric setting the optimal policy in terms of average delay and the individual optimal policy for jobs arriving at a particular primary server coincide (hence, a Nash solution). This is not the case when one considers asymmetric secondary servers. Figures 2 and 3 illustrate this phenomenon. Consider a system of 2 queues and 2 M/M/1 servers where server capacities are different, i.e., $E[\tau_1] \neq E[\tau_2]$. Figure 2 and 3 illustrate the delay performance versus $E[\tau_2]$ while keeping $E[\tau_1] = 1$ fixed. Both figures provide the delay of each stream under no sharing $D_i, i = 1, 2$, their average, D_{avg}, and the delay, D_{shared}, of a sharing policy which distributes all jobs in a stochastic sense (the average delay of both streams in this case are equal). Notice that we intuitively expect that the policy that stochastically balances the waiting time of each piece minimizes the average delay, i.e., $D_{shared} \leq D_{avg}$, as verified in Figures 2 and 3. Both figures illustrate that when $E[\tau_1]$ and $E[\tau_2]$ are roughly of the same order the policy that achieves the minimum average delay also results in delay improvements for each stream, i.e., $D_{shared} \leq D_i, i = 1, 2$. This implies that there exists a natural incentive for users to share (asymmetric) resources so long as the resources are of the same order. These figures, on the other hand, exhibit different behavior as $E[\tau_2]$ grows (resource of user 2 becomes significantly worse than that of user 1). When service time has a low variance, as shown in Figure 2, both users see a delay improvement for all stabilizable service parameters (hence, an incentive to share). This is in contrast to the case where service times have a large variance. This case is shown in Figure 3, where for high value of $E[\tau_2] > 4E[\tau_1]$, user 1 sees a degradation in performance, i.e., $E(D_1) < E(D_{shared})$. In other words, in this case the average delay improvement comes only at the cost of the session delay for jobs of user 1, creating a disincentive for user 1 to share service with user 2!

Overall, these figures show that in the asymmetric extension of the model, users with fast servers might, or might not, have an incentive to enter the sharing coalition. This brings up many interesting issues in coalition and game theory as they apply to resource sharing problems with respect to delay performance.

Acknowlegements

The author would like to thank Prof. Cruz for many useful discussions and feedback and Mr. E. Ardestani who helped with the figures. The author would also like to thank the reviewers for their valuable feedback. The recommendations from the anonymous reviewers have helped us greatly in revising the manuscript.

REFERENCES

[1] S. K. Agarwal, M. Laifenfeld, A. Trachtenberg, and M. Alanyali. Using bandwidth sharing to fairly overcome channel asymmetry. In *Proceedings of Information Theory and Application, San Diego*, Feb 2006.

[2] S. Asmussen. *Applied Probability and Queues*. Springer, 2000.

[3] F. Baccelli. Two parallel queues created by arrivals with two demands; the m/g/2 symmetrical case. Technical report, 1985.

[4] F. Baccelli and Z. Liu. On the execution of parallel programs on multiprocessor systems: a queueing theoretic approach. *J. ACM*, 1990.

[5] F. Baccelli, W.A. Massey, and D. Towsley. Acyclic forkjoin queuing networks. *J. ACM*, 1989.

[6] G. Barrenechea, B. Beferull-Lozano, A. Verma, P. Luigi Dragotti, and M. Vetterli. Multiple description source coding and diversity routing: A joint source channel coding approach to real-time services over dense networks. *Proc 13th Int. Packet Video Workshop*, 2003.

[7] B. B. Bhattacharyya. Reverse submartingale and some functions of order statistics. *Ann. Math. Statist.*, 41:2155–2157, 1970.

[8] J. Chakareski, S. Han, and B. Girod. Layered coding vs. multiple descriptions for video streaming over multiple paths. In *Proceedings of ACM Multidmedia*, November 2003.

[9] C.S. Chang, R. Nelson, and D.D. Yao. Scheduling parallel processors: structural properties and optimal policies. *Math. Comput. Model*, 23, 1994.

[10] H. A. David and H. N. Nagaraja. *Order Statistics*. Wiley, 2003.

[11] M. Harchol-Balter. Task assignment with unknown duration. *Journal of the ACM*, December 2002.

[12] http://www.mushroomnetworks.com. APX10 access point aggregator, for the enlightened wireless data user. Technical report.

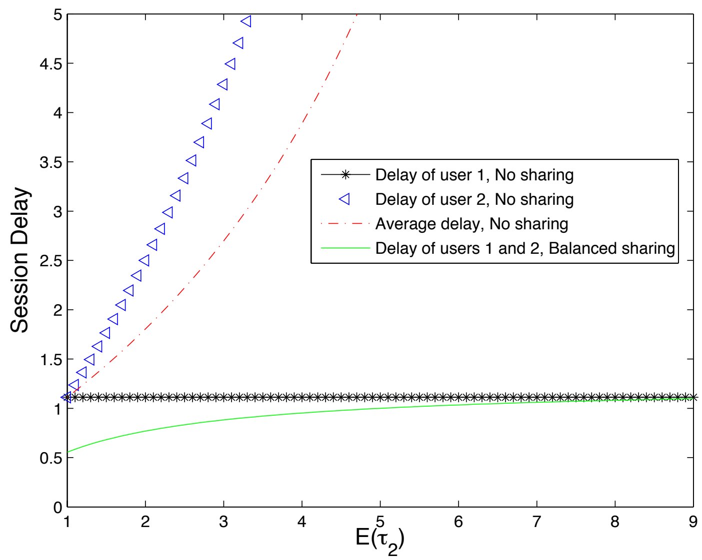

Fig. 2. Session Delay versus $E(\tau_2)$ when $E(\tau_1) = 1$ and variance is 2 (low)

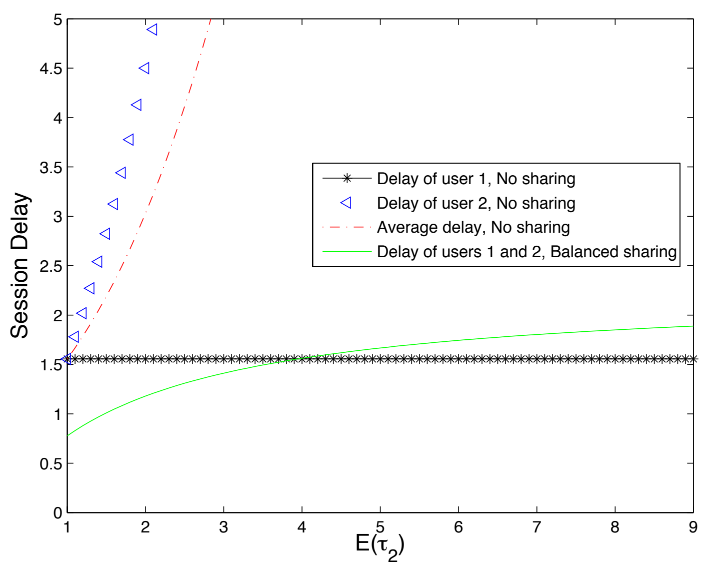

Fig. 3. Session Delay versus $E(\tau_2)$ when $E(\tau_1) = 1$ and variance is 10 (high)

[13] T. Javidi. Cooperation and resource sharing in data networks: A delay perspective. In *Proceedings of Allerton Conference*, 2006.
[14] P. Konstantopoulos and J. Walrand. Stationarity and stability of fork/join networks. *J. Appl. Probab*, 1989.
[15] J. N. Laneman, D. N. C. Tse, and G. W. Wornell. Cooperative diversity in wireless networks: Efficient protocols and outage behavior. *IEEE Transactions on Information Theory*, 50(12).
[16] A. M. Makowski and R. Nelson. An optimal scheduling policy for fork/join queues. In *Proceedings of the 32nd IEEE Conference on Decision and Control*, 1993.
[17] S. Mao. Multi-path routing for multiple description video over wireless ad hoc networks. In *Proceedings of IEEE Infocom*, 2005.
[18] S. Mao, S. Lin, Y. Wang, S. Panwar, and Y. Li. Multipath video transport over ad hoc networks. *IEEE Wireless Communications Magazine*, 12, 2005.
[19] A. W. Marshall and I. Olkin. *Inequalities: Theory of Majorization and Its Applications*. Academic Press, 1979.
[20] A. Muller and D. Stoyan. *Comparison Methods for Stochastic Models and Risks*. Wiley, 2002.
[21] R. Nelson and A.N. Tantawi. Approximate analysis of fork/join synchronization in parallel queues. *IEEE Trans. Comput*, 1988.
[22] R. Nelson, D. Towsley, and A.N. Tantawi. Performance analysis of parallel processing systems. *IEEE Trans. Software Eng.*, 1988.
[23] E. Setton, Y. Liang, and B. Girod. Adpative multiple description video streaming over multiple channels with active probing. In *Proceedings of IEEE ICME*, July 2003.
[24] C. Xia, G. Michailidis, and N. Bambos. Dynamic on-line task scheduling on parallel processors. *Performance Evaluation*, 46(2-3), October 2001.