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

Tara Javidi
Department of Electrical and Computer Engineering
University of California, San Diego
tara@ece.ucsd.edu

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 the network performance. From the perspective of any one user, when ignoring the interaction among users, all such schemes reduce to various forms of providing parallelism and, hence, increased throughput. In this paper, we argue that focusing on parallelism is by no means sufficient as it ignores the existence of many users with potentially similar strategies.

In this paper, we illustrate the issue of resource sharing in the above context via a multi-queue multi-server problem. Our proposed model is not realistic since it ignores the overhead inefficiencies, yet it does capture the trade-off between parallelism and reassembly/synchronization delay to a large extent. We use this model to provide analytical results in a special case of homogeneous users and servers. Furthermore, we prove the robustness of a certain locally optimal strategy to non-cooperation in a Nash equilibrium strategy context.

I. INTRODUCTION

With the growth of networking technologies, scheduling and resource allocation have become important topics of interest when considering network performance. Due to size and the decentralized nature of networks today, many applications and resource allocation schemes rely heavily on resource pooling and sharing to overcome the inherent limitations of the system. Among such applications, one can mention 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, for example. This application is used to facilitate a remote access to the files which reside in ones “home” machine. Such an access can potentially be negatively affected by the asymmetry in upload/download speed. To overcome the limitation caused by a low upload speed, content distribution solutions rely on cooperation among many users such that the distribution of ones’ files is distributed over many home machines and a remote access would use parallel upload of these distributed pieces. It is, then, intuitive to notice an increased peak rate. This problem has been introduced in [2], where strategy-proof sharing policies have been studied and analyzed. Furthermore, it is showed that such policies result in a fair sharing of the “additional unused” bandwidth of the network. We believe that the very notion of throughput as a long run average quantity prevents any such study to capture the main advantage of content distribution solutions, which is the increase in peak rate, or more precisely delay improvements. In this paper, we provide a simple queueing theoretic approach to capture the effect of content distribution with respect to delay.

Similarly, delay improvements can be obtained in the context of multi-description coding and multi-path routing. Multi-description coding combined with multi-path routing has been proposed to improve quality of service and delay profile for multi-media applications. The main idea behind multi-description coding of a multi-media file is to break and encode it into many descriptions (pieces with some redundancy) such that reception of each description by itself is sufficient for low-quality reproduction of the file and a larger number of received descriptions result in a higher quality reproduction [6]. Many authors have suggested disjoint (and simultaneous) multi-path routing of the descriptions of a file to improve session delay. Most studies, though, have focused on the impact of multi-path routing on one particular user, while assuming a fixed network model (see IEEE Wireless Communications Magazine, Special Issue on Advances in Wireless Video, June 2005). It is not surprising that such assumption exhibits significant improvements due to an increased parallelism. 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. In this paper, we use a simple queueing model to show that the parallelism advantage should be balanced against a potential increase in load due to added redundancy as well as an excessive reassembly delay (due to backlog fluctuations) and synchronization delay. Notice that in this paper, we do not consider cases where multi-path routing does also provide a load balancing benefit such as those in [6].

Similar queue models can be used to analyze the benefits of community networking and access point sharing. Many new schemes of access point 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. Ideally, due a statistical multiplexing gain, an increase in peak rate is possible. On the other hand, such an advantage must be balanced against a drop in throughput caused by increase contention levels. Again our paper, we believe, provides a first step towards delay analysis in such systems.

The above three systems can all be modeled as a multi-queue multi-server problem with stochastic job distributions, where arriving jobs can be divided into pieces and be sent to many queues and servers to take advantage of parallelism. In this queueing model (also known as a fork and join queue model), each job
corresponds to a session, multi-media file, etc. We notice that the most important advantage of resource sharing is related to reduction in response time as a result of statistical multiplexing, since in a throughput increase is not possible. Unfortunately, delay analysis of multi-server problems queuing is much more complicated than throughput analysis. This is exactly why, we believe, analysis of the delay improvements in systems above have not received sufficient treatment. In this paper, we attempt to tackled the issue of delay performance of sharing policies from three distinct (but related) angles:

1. In the absence of backlog information, what is the optimal (with respect to application layer delay) sharing policy among all policies that result in a symmetric treatment of servers?
2. How does such an optimal symmetric sharing policy compare with an exclusive model where each stream of jobs is assigned to one particular server exclusively?
3. How robust is such an optimal symmetric sharing policy to non-cooperative behavior?

The remainder of the paper is organized as follows. In Section II, we provide a simple queueing model which captures some of salient feature of the problem in case of homogenous users and servers. In Section II, we also discuss the prior work. In Section III, we show that the policy that divides and distributes large jobs in equal size pieces, \( A^\ast \), is optimal among all policies that share servers symmetrically. In Section IV we show for small enough of redundancy/overhead such policy outperforms a non-sharing policy, where each stream of jobs is exclusively assigned to a server. In Section V, we show that the aforementioned optimal symmetric sharing policy, \( A^\ast \), is also a Nash equilibrium for the non-cooperative job distribution game. In other words, we show that policy \( A^\ast \) not only improves the delay performance of the system for all users, but also exhibits robustness to non-cooperation. In Section VI, we summarize and conclude the paper.

II. PROBLEM FORMULATION

A. A Simple Queueing Model

In order to analyze the delay performance of possible sharing schemes we use a simple queueing model, consisting of \( n \) queues and \( n \) servers. As we will see this problem is related to the classical fork/join queueing problems.

We assume that \( n \) stream of jobs arrive at \( n \) primary servers/queues according to \( n \) independent Poisson processes each with rate \( \lambda \). The size of a given job at stream \( i \), \( \tau_i \), is drawn from a general distribution \( G \) with mean \( 1/\mu \). These primary servers are responsible to forward the ”small” jobs to the secondary queues or divide and distribute the ”large” jobs across the servers. Jobs are queued at the secondary queue if 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, and without any knowledge of the state of the system. Secondary queues are all served at fixed unit rate and jobs leave the system after completing service at secondary queues. There is a redundancy factor \( 1 < \beta \) associated with the sharing overhead. We consider a class of (time invariant) job distribution policies as follows:

- Jobs of stream \( i \) whose size are smaller than a threshold, \( \tau_0 \), are simply forwarded to secondary server \( i \)
- Jobs whose size is greater than \( \tau_0 \) are broken in \( n \) pieces and distributed to secondary servers according to a fixed distribution matrix \( A = [\alpha^1, \alpha^2, \ldots, \alpha^n] \), where \( A' \) is a stochastic matrix.
- Forwarding and division of jobs are done in a time invariant manner without knowledge of secondary queue backlogs.

We are interested to find the ”optimal” \( A \) for a given \( \tau_0, \lambda, G, \) and \( \beta \), where the performance measure of interest is the expected session delay, also known as application layer delay.

**Definition 1:** The session delay of a job is the delay the job sees from the moment it arrives at the system until its last piece finishes service at a secondary server.

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

\[
D_{\tau,i}(A) = \begin{cases} 
\tau + W_i(A) & \tau < \tau_0 \\
\max_j \{\alpha^1_j \beta \tau + W_j(A)\} & \tau > \tau_0
\end{cases}
\]

where \( W_i(A), i = 1, \ldots, n \), is the stationary waiting time at secondary queue \( i \).

We are interested in minimizing \( E(D_{\tau,i}(A)) \) for a given \( \tau \) and \( \tau \) over choices of distribution matrix (the expectation is taken over the random arrivals and job sizes associated with all jobs but the last one). For the reasons discussed earlier we address this problem in the context of problems (P1)-(P3). Before introducing the problems, we need the following definition:

**Definition 2:** A policy \( A \) is called symmetric strict sharing (SSS) iff it divides the jobs in such a way that 1) all secondary servers receive a positive load from each primary server and 2) the division vectors \( \alpha^i \) for primary queues \( i, i = 1, \ldots, n \), are permutations of each other. We denote the class of such policies by \( A_\alpha \).

It is easy to see that \( A_\alpha \) is the classes of all matrices \( A \) such that \( A \) is a cyclic doubly stochastic matrix whose elements are strictly positive.

**P1** Show that for any job of given size \( \tau \), equal sharing matrix \( A^\ast = [1/\mu]_{n \times m} \) minimizes delay among all symmetric strict sharing policies, i.e. for all \( \tau \),

\[
A^\ast = \arg \inf_{A \in A_\alpha} E(D_{\tau,i}(A)).
\]

**P2** Show that for small enough of redundancy factor, \( \beta \), one can always find an appropriate threshold, \( \tau_0 \) such that the distribution policy \( A^\ast \) outperforms the non-cooperative policy of serving jobs of each stream exclusively on a particular server.

**P3** Show that given a level of autonomy among users (which is the only reasonable assumption in scenarios of our interest), the
optimal distribution policy $A^*$ is also a Nash equilibrium of the non-cooperative game.

Note that restricting attention to $A_s$ ensures that for $\beta = 1$ the load of each server under sharing remains the same as an exclusive allocation. In other words, in case of server ownership, cooperation does not cause a load increase on any user’s resource. From the definition of $A_s$, we can simplify notations by replacing $A$ with its first column. For instance, we can write $D_\tau(\alpha) := D_\tau,i(A(\alpha))$, where $A(\alpha)$ is a cyclic matrix whose columns are permutations of $\alpha$.

B. Related Work

The first problem is almost identical to the one studied in [8], [14], and [7], with one major difference that in these papers servers’ backlog information is available to a centralized controller. We note that the seeming contradictory result with those presented in [14] is not surprising when one considers the value of state information. The sequential policy in [14] is different from the policy that forwards the jobs exclusively to a particular server, since the sequential policy in [14] can join any of the secondary queues with low backlog. 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. It’s known that very different results are obtained when considering different assumptions on job size distribution, $G$. For instance, when the queue information is available and when the job size distribution follows an exponential model, joining the shortest queue is known to be optimal [18], while in case of pareto job size distributions, a size-based assignment is advantageous [11]. (For a complete discussion on task assignment problems with various assumptions, see [10] and the references therein.) 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, albeit at the cost of added redundancy/overhead. The second difference is more philosophical.

In this paper, we restrict our attention to a subset of policies, we call symmetric sharing. Unlike the traditional task assignment problems where all servers are owned and managed by an entity whose interests are 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 problem 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 non-cooperative and selfish behavior. We concretize this matter, when addressing P3, via the notion of Nash equilibrium.

After identifying the policy that divides and distributes jobs (or a subset of them that are considered “too large”) in equal size pieces as the optimal policy among the class of SSS policies, we establish a simple result comparing such policy with a non-sharing policy in which each stream of jobs is allocated exclusively to one particular server. Our result depends on continuity of delay with respect to parameters $\tau_0$ and $\beta$ to arrive at existence results which show that for any admissible load, there always exist a level of redundancy below which all equal sharing policy (for a careful choice of $\tau_0$) outperforms exclusive allocation of servers and vice versa. This result are a special case of the performance analysis of a classical fork/join queuing problem ([3], [4], [5], [13], [16], and [17]) where the notion of redundancy and the optimality of all equal sharing provides us with much more structure. The extension of results in [3] and [4] (much stronger than ours) to this instant remains to be a topic of future study.

Notation: Throughout the paper, we use $[1/n]_\times$ 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.

III. Analysis of Problem P1

As mentioned before, in this Section, we restrict our attention to distribution matrices $A \in A_s$. Theorem 1 below provides the first result of our paper.

Theorem 1: Assume $\tau_0, G$, and $\lambda$ are given and fixed. For any given job of size $\tau$ of stream $i$, and among all $A = A(\alpha)$ such that $A \in A_s$, the delay is minimized at $A^* = [1/n]_\times$, i.e.

\[
E(D_\tau(1/n)) \leq E(D_\tau(\alpha))
\]

where $D_\tau(1/n)$ is a shorthand for $D_\tau(1/n, 1/n, \ldots, 1/n)$.

Proof: Remember that

\[
E(D_\tau(\alpha)) = \begin{cases} 
\tau + E(W_i(\alpha)) & \tau \geq \tau_0 \\
\max_j \{\alpha_j \beta \tau + W_j(\alpha)\} & \tau > \tau_0 
\end{cases}
\]

\[
E(D_\tau(1/n)) = \begin{cases} 
\tau + E(W_i(\alpha)) & \tau \geq \tau_0 \\
\beta \tau/n + E(\max_j \{W_j(1/n)\}) & \tau > \tau_0
\end{cases}
\]

Note that for the rest of this proof, we can restrict attention to $\beta = 1$ without any loss of generality. This is because that all symmetric sharing policies incur similar redundancy overhead.

We use the following fact.

Fact 1: For any $\alpha$, we have

\[
W_i(1/n) \leq W_i(\alpha)
\]

\[
(W_1(1/n), \ldots, W_n(1/n)) \leq (W_1(\alpha), \ldots, W_n(\alpha))
\]

where $\leq$ denotes the usual stochastic order defined on space of random vectors.

This fact is a direct consequence of Theorems 6.3.5 and 3.3.7 of [1]. All we need to notice is that a job arriving at secondary queue $i$ has a random size $\tau_0$, which is equal to $\alpha_i \tau$, $i = 1, \ldots, n$ with equal probability $1/n$. It is then easy to verify that

\[
\tau_{\alpha} \leq \tau/n.
\]

Now using Fact 1, we arrive at the following lemma.

Lemma 1: For any $\alpha$, we have

\[
E(\max \{\alpha_j \tau + W_j(1/n)\}) \leq E(\max \{\alpha_j \tau + W_j(\alpha)\})
\]

The proof of this lemma is given in the appendix. Furthermore, we show

Lemma 2: For any $\alpha$, we have

\[
E(\max \{\tau/n + W_j(1/n)\}) \leq E(\max \{\alpha_j \tau + W_j(1/n)\})
\]

The proof of this lemmas is also given in the appendix.
IV. ANALYSIS OF PROBLEM P2

We are interested in comparing \( E(D_\tau(1/n)) \) with \( E(D_\tau^{ns}) \), where the superscript of \( ns \) refers to a non-sharing policy which allocates all jobs of stream \( i \) to server \( i \). The main result of this section is given by the following theorem:

**Theorem 2:** Given a load \( \rho < 1 \) for the non-sharing system, there exists a threshold \( \tau_0 \) and small enough \( \beta \), under which the all-equal sharing policy outperforms the non-sharing policy. Conversely, there exists a large enough \( \beta \) for which sharing is inferior.

**Proof:**

To start, let us fix \( \tau_0 \). We now compare the average length of a secondary queue under both policies. In other words, we aim to compute \( E(W(1/n)) \) and \( E(W^{ns}) \). To do so, we notice that each secondary queue \( i \) now is an M/G/1 system; we then use the Pollaczek-Khinchin formula to calculate the expected \( E \) waiting time at each secondary queue. Using this we arrive at

\[
E(W^{ns}) = \frac{\lambda E(\tau^2)}{2(1-\rho)}. \]

All we need, now, is to calculate arrival rate \( \lambda' \) and the first and second moments, \( E(\tau') \) and \( E(\tau'^2) \), of job size distributions at secondary queue \( i \). It is easy to verify that

\[
\lambda' = \lambda(n_0 + n(1-q_0)) = \lambda(n - (n-1)q_0)
\]

\[
E(\tau') = \frac{E(\tau)}{\lambda \tau} \frac{(n-1)q_0}{\lambda - (n-1)q_0 s_1}
\]

\[
E(\tau'^2) = \frac{E(\tau'^2)(1 + q_0 s_1)}{\lambda \tau^2} \frac{(n-1)q_0 s_2}{\lambda - (n-1)q_0 s_1}
\]

where \( q_0 = G(\tau_0), s_1 = \frac{E(\tau|x<n)}{E(\tau)}, \) and \( s_2 = \frac{E(\tau'^2|x<n)}{E(\tau'^2)} \).

As a result, we have

\[
E(W_j(1/n)) = \frac{\lambda E(\tau^2)}{2(1-\rho)} \frac{(1-\rho)(\beta^2 - \beta^2 q_0 s_1 + nq_0 s_2)}{1 - \rho(\beta - \beta q_0 s_1 + q_0 s_1)}
\]

**Case 1:** \( \tau < \tau_0 \)

\[
E(D_\tau(1/n)) = \tau + E(W^{ns}) = \tau + \frac{\lambda E(\tau^2)}{2(1-\rho)}
\]

while

\[
E(D_\tau^{ns}) = \tau + \frac{\lambda E(\tau^2)}{2(1-\rho)} (1 - \rho)(\beta^2 - \beta^2 q_0 s_1 + nq_0 s_2)
\]

**Case 2:** \( \tau > \tau_0 \)

Similarly, from P-K formula, we have

\[
E(D_\tau(1/n)) = \tau/n + E(W(1/n)) \leq \tau/n + nE(W)
\]

\[
\leq \tau/n + \frac{\lambda E(\tau^2)}{2(1-\rho)} (1 - \rho)(\beta^2 - \beta^2 q_0 s_1 + nq_0 s_2)
\]

In other words, we have shown that for all \( \tau \) and any given load \( \rho < 1 \), \( E(D_\tau(1/n)) \) is always bounded continuous function at an epsilon neighborhood of \( (\beta, \tau_0) = (1, 0) \). Furthermore, we have shown that when \( \beta = 1 \), and \( \tau_0 = 0 \), \( E(D_\tau(1/n)) < E(D_\tau^{ns}) \). From this we have the first part of the theorem.

So far we have shown that there always exists a small enough redundancy and careful choice of \( \tau_0 \) such that sharing outperforms non-sharing policies. It is easy to construct heavy traffic examples under which the increase in network load, caused by redundancy \( \beta \), makes sharing inferior to sharing. To prove the second part of the theorem, we notice that for any \( \rho \) there exists a large enough \( \beta \), such that \( \rho' \) can be made greater than 1. This corresponds to infinite delay for both small and large size jobs, independent of threshold \( \tau_0 \).

So far we have only presented existence result (in terms of parameters) for each system to outperform the other. Figures IV and IV, show the resulting average delays in simulations of M/M/1 queues versus redundancy for various choices of threshold \( \tau_0 \), when \( \lambda = 0.4 \) and 0.9, respectively. One can observe the complexity of the dependency on \( \beta \) and \( \tau_0 \). Note that, for instance at low redundancy, one needs to choose low thresholds to improve the waiting time at queues while at higher redundancy, higher threshold \( \tau_0 \) help remedy the increase in load. An interesting area of future research is an exact characterization of regimes where each policy is optimal. We believe results from fork/join queueing will be crucial [3], [4]. Furthermore, we notice that in our existence proof, the provided \( \beta \) and \( \tau_0 \) are appropriately chosen to guarantee similar performance for both small and large jobs. This is a scenario where individual and social optimality coincide. This proof does not answer an interesting question regarding the possibility of conflicting interests. In other words, we have not considered the possibility...
of cases under which small jobs “prefer” one policy while large jobs “prefer” the other.

In the next section we show that when sharing outperforms exclusive server allocation and given users’ adherence to an all-equal sharing policy, no user can decrease her delay unilaterally.

V. ANALYSIS OF PROBLEM P3

In this section, we show that for small $\beta$ and carefully chosen $\tau_0$ (when all-equal sharing outperforms the non-sharing) all equal sharing also constitutes a Nash equilibrium. In other words, we show that given that all users, but one, follow an all-equal splitting rule (and in the absence of queue backlog information), the average delay of that user can only increase if she chooses to deviate from all-equal splitting rule. This is articulated in the following theorem:

**Theorem 3:** Without loss of generality consider user $i$, then

$$D_{r,i}(\alpha^+, \alpha^{-i}) \leq D_{r,i}(\alpha^i, \alpha^{-i})$$

where $\alpha^i$ is any splitting vector (elements add up to 1), $\alpha^+ = [1/n]_{n \times 1}$ and $\alpha^{-i} = [1/n]_{n \times n-1}$.

**Proof:**

- Recall that

$$E(D_{r,i}(\alpha^i, \alpha^{-i})) = E\left(\max_j \left\{ \alpha_j \tau + W_j(\alpha, \alpha^{-i}) \right\} \right)$$

$$E(D_{r,i}(\alpha^+, \alpha^{-i})) = E\left(\max_j \{ \tau/n + W_j(1/n) \} \right)$$

Now we use the following lemma to arrive at the assertion of the theorem. Notice that we denote jobs for which $\tau \geq \tau_0 (\tau < \tau_0)$ as large (small) jobs.

**Lemma 3:** Consider an arbitrary realization of the sequence of inter-arrival times and job sizes associated with all “large” jobs arriving at primary queue $i$. Define $f(\alpha^i) = E\left(\max_j \{ \alpha_j \tau + W_j(\alpha, \alpha^{-i}) \} \right)$, where the expectation is taken with respect to the job sizes and arrivals to all other queues as well as the small jobs arriving at queue $i$. Then $f(\alpha)$ is symmetric and convex in $\alpha$.

From this lemma we have that for any given realization of the sequence of inter-arrival and job sizes of “large” jobs into queue $i$, $E(D_{r,i}(\alpha^i, \alpha^{-i}))$ is symmetric and convex in $\alpha$. This means that we have

$$D_{r,i}(\alpha^+, \alpha^{-i}) \leq D_{r,i}(\alpha^i, \alpha^{-i})$$

hence, the assertion of the theorem.

Again proof of Lemma 3 is given in the appendix.

VI. FUTURE WORK

We have a long way in fully characterizing the delay improvement in this system. In the current paper, we have showed that there exists a parameter space under which sharing servers in a simple-to-implement way is beneficial and there are conditions under which is not. We hope to strengthen this result by taking advantage of analysis techniques applied to fork/join queuing problems. In addition, we would like to extend the current result to other queueing disciplines.

Acknowledgement

This research is in part funded by Center for Networked Systems research grant CNS05009.

REFERENCES


APPENDIX

I. PROOF OF LEMMA 1

**Lemma 1:** For any $\alpha$, we have

$$E(\max_j \{ \alpha_j \tau + W_j(1/n) \}) \leq E(\max_j \alpha_j \tau + W_j(\alpha))$$
Proof:
We first use the following fact from [1] (Theorem 3.3.1, page 94):

Fact 2: If $\mathbf{X} \leq_{st} \mathbf{Y}$ and $g : \mathbb{R}^n \rightarrow \mathbb{R}^k$ is increasing, then $g(\mathbf{X}) \leq_{st} g(\mathbf{Y})$.

Now, given a fix $\tau$, function $g(x) = \max_j \{ \alpha_j \tau + x_j \}$ is an increasing function. Hence, from Fact 1 and the definition of stochastic order for finite mean variables, we have the theorem. ■

II. PROOF OF LEMMA 2

Lemma 2: For any $\alpha$, we have

$E(\max \{ \tau/n + W_j(1/n) \}) \leq E(\max \{ \alpha_j \tau + W_j(1/n) \})$

Proof:
In order to prove this lemma, we first prove the following claim.

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

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

Then

$$\max_i \bar{f}(X_i) \leq_{icx} \max_i f_i(X_i).$$

Recall that $n$ random variables are called exchangeable if the joint distribution of of the random vector is permutation invariant and that

$$X \leq_{icx} Y \iff Ef(X) \leq Ef(Y) \quad \forall f \text{ increasing, convex.}$$

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

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

$$E(Y | \max_i \bar{f}(X_i))$$

$$= E(X_1, \ldots, X_n | \max_i \bar{f}(X_i)) \left(E(Y | X_1, \ldots, X_n, \max_i \bar{f}(X_i)) \right)$$

$$= E(X_1, \ldots, X_n | \max_i \bar{f}(X_i)) \left(E(Y | X_1, \ldots, X_n) \right)$$

$$= E(X_1, \ldots, X_n | \max_i \bar{f}(X_i)) \left(\max_i f_\pi(i) \right)$$

$$= E(X_1, \ldots, X_n | \max_i \bar{f}(X_i)) \left(\frac{1}{n!} \sum_{\pi \in \mathcal{P}} \max_i f_\pi(i) \right)$$

$$\geq \frac{1}{n!} E(X_1, \ldots, X_n | \max_i \bar{f}(X_i)) \left(\max_i \sum_{\pi \in \mathcal{P}} f_\pi(i) \right)$$

$$= \frac{1}{n!} E(X_1, \ldots, X_n | \max_i \bar{f}(X_i)) \left( (n-1)! \sum_i f_i(X_i) \right)$$

$$= E(X_1, \ldots, X_n | \max_i \bar{f}(X_i)) \left( \max_i \bar{f}(X_i) \right)$$

$$= \max_i \bar{f}(X_i).$$

Noting the exchangeability of $W_j(1/n)$, we put $f_j(x) = \alpha_j \tau + x$ and $\bar{f}(x) = \tau/n + x$. Using (1), we now have the assertion of the lemma. ■

III. PROOF OF LEMMA 3

Lemma 3: Consider an arbitrary realization of the sequence of inter-arrival times and job sizes associated with “large” jobs arriving at primary queue $i$. Define $f(\alpha^i) = E \{ \max_j (\alpha_j \tau + W_j(\alpha^i, \tau^{n-1})) \}$, where the expectation is taken with respect to the job sizes and arrivals to all other queues as well as the small jobs arriving at queue $i$. Then $f(\alpha)$ is symmetric and convex in $\alpha$.

Proof:
Without loss of generality, put $i = 1$. To prove the lemma we use the following fact (Proposition B.4 in [15]):

Fact 3: Let $X_1, \ldots, X_n$ be exchangeable random variables. Let $\Phi(x, \alpha) = \phi(w(x_1, \alpha_1), \ldots, w(x_n, \alpha_n))$ where $\phi$ is symmetric, increasing, and convex (on $\mathbb{R}^n$), and for each fixed $z$, $w(z, \cdot)$ is convex (on $\mathbb{R}$). With the appropriate measurability,

$$\psi(\alpha) = E\Phi(X; \alpha)$$

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

Now consider an arbitrary realization of the sequence of inter-arrival times and job sizes associated with “large” jobs arriving at primary queue $i$. Due to the Poisson assumption on arrival process and the independence between inter-arrival times and job sizes, we can, in general, consider the arrival to each primary queue as a superposition of two Poisson arrival streams with rates $\lambda_0 \lambda$ and $(1 - \lambda_0) \lambda$ [9].

Now consider the secondary queues. Arrival process into secondary queue $j$, $j = 1, 2, \ldots, n$, now, consists of super-positioning of three Poisson streams whose job sizes we denote by

- $\tau^T$: These are small jobs arriving at primary queue $j$. Since they are small, they are not divided among queues but are directly forwarded to secondary queue $j$. The size and the inter-arrival times of these jobs are exchangeable across queues (due to homogeneity of servers/queues).
- $\tau^L$: These are pieces of large jobs that have arrived at primary queues but primary queue 1. These jobs are split equally among all secondary queues; their arrival times and sizes are identical across queues.

- $\tau^L$: These are pieces of large jobs arriving at primary queue 1 which have been forwarded to secondary queue $j$. Their distribution depends on $\alpha^1_j$ and their realization is assumed fixed in this lemma.

Now notice that across all secondary queues, $\tau^T$, $\tau^L$ and their corresponding inter-arrival times $B^T_j$ and $B^L_j$ are exchangeable random variables. Furthermore, each secondary queue’s waiting $W_j$ is a function of these exchangeable random variables as well as the specific sizes of large jobs of primary queue 1, hence, a function of $\alpha_j$. In other words, we can put $\alpha^{j} \tau + W_j(\alpha^j, \tau^{n-1}) = w(\{X\}, \alpha_j)$, where $\{X\}$ is a short hand for the vector of random variables associated with all random events in the system, i.e. $\tau^j, \tau^L, B^T_j$ and $B^L_j$ for all arrived jobs but those of the large jobs arrived at primary queue 1. Note that for given fixed values of $\tau^T, \tau^L, B^T_j$ and $B^L_j$, $w$ is a convex in $\alpha_j$ (see section 6.3 in [1]). On the other hand, notice that $\phi(y) = \max(y)$ is convex, symmetric and increasing. Now taking expectation sequentially on each variable of sequence $\{X\}$ and using Fact 3, we arrive at the assertion of the lemma. ■