Cooperation and Resource Sharing in Data 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 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. Although, our proposed model is not realistic in that it ignores the overhead inefficiencies, it does capture the trade-off between parallelism and reassembly/synchronization delay to a larger 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 the 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 one's "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 [1], 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-desrciption 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 the reception of each description by itself is sufficient for a low-quality reproduction of the file and more and more received descriptions result in a higher and higher quality of reproduction [5]. 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 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 [5].

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

This work was supported by a grant from Center for Networked Systems

T. Javidi is with Faculty of Electrical and Computer Engineering, University of California, San Diego, 92093-0407 tara@ece.ucsd.edu

started to appear as popular solutions across communities (for one example, see [10]). 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 off against a drop in throughput caused by contention. 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, even when throughput increase is impossible. Unfortunately, despite its intuitive nature, delay analysis of multi-server problems queuing is much more complicated than throughput analysis. 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 queuing 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 (in a stochastic sense) pieces, A^*, is optimal among all policies that result in a symmetric loading of servers. This optimality is a strong increasing convex stochastic sense. In Section IV, we show that 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^*, is also a Nash equilibrium for the non-cooperative job distribution game. In other words, we show that policy A^* 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.

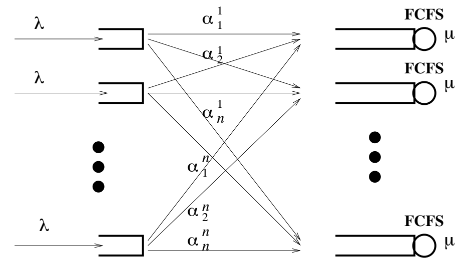

Fig. 1. A simple queueing model for resource sharing among homogeneous users

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 λ. These primary servers are responsible to divide and distribute the 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. The service time associated with any given (complete) job at a secondary queue i is a random variable τ whose distribution is G with mean $1/\mu$. Secondary queues are assumed to be independent of each other and of arrivals. The service time of any piece of a job (as a result of division and distribution at a primary queue) is linear in the splitting ratio. Each (piece of a) job leaves the system after completing service at a secondary queues. We consider a class of (time invariant) job distribution policies as follows:

- Jobs are broken in n pieces and distributed to secondary servers according to a fixed distribution matrix $A = \left[\underline{\alpha}^1\underline{\alpha}^2\cdots\underline{\alpha}^n\right]$ whose transpose, A', is a stochastic matrix.
- Splitting factor α^i_j is such that the service time of a piece of the job coming from primary queue i at the secondary queue j is $\alpha^i_j\tau_j$, where $\tau_j \sim G$. Furthermore, we assume τ_i and τ_j are exchangeable variables (see Definition 3 in the Appendix).
- 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 choice of λ and G, 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 steady state delay in the stream i (in steady state) is given by

$$D_i(A) = \max_j \left\{ \alpha_j^i \tau_j + W_j(A) \right\} \tag{1}$$

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

We are interested in minimizing $E(D_i(A))$ for a given i over choices of distribution matrix (the expectation is taken over the random arrivals and job sizes). For the reasons discussed earlier we address this problem in the context of problems (P1)-(P3). But 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 α^i for primary queues $i, i = 1, \ldots, n$ are permutations of each other. We denote the class of such policies by $\mathcal{A}_s$.

It is easy to see that $\mathcal{A}_s$ is the clases of all matrices A such that A is a cyclic doubly stochastic matrix whose elements are strictly positive.

Now we are ready to introduce the following problems:

P1: Show that equal sharing matrix $A^* = [1/n]_{n \times m}$ minimizes delay among all symmetric strict sharing policies, i.e.

$$A^* = \arg \inf_{A \in \mathcal{A}_s} E(D_i(A)). \tag{2}$$

P2: Show that the distribution policy A^* 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 $\mathcal{A}_s$ ensures that the load of each server under sharing remains the same as that under the exclusive allocation. In other words, in case of server ownership, cooperation does not cause a load increase on any user's resource.

B. Related Work

The first problem, (P1), is almost identical to the one studied in [7], [12], and [18], 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 [12] is not surprising when one considers the value of state information. The sequential policy in [12] is different from the policy that forwards the jobs exclusively to a particular server, since the sequential policy in [12] can join *any* of the secondary queues. Unlike the traditional fork/join 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. As a result, our very definition of non-sharing policy is more restrictive than the definition of sequential scheduling in [7] and [12]. We will elaborate on this further in the next paragraph.

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 is known that the properties of the optimal assignment strategy varies substantially 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 [17], while in case of pareto job size distributions, a size-based assignment is advantageous [9]. (For a complete discussion on task assignment problems with various assumptions, see [8] 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. The second difference is more philosophical. In this paper, we restrict our attention to a subset of policies, we call *symmetric sharing*. As mentioned earlier, we are interested in applications where there is an association between a particular (secondary) server and a particular stream of jobs. This not only has implications on what policies can be assumed as acceptable (as described in case of non-sharing policies vs. sequential policies) but also results in interesting results in the context of non-cooperative behavior. In this paper, 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 may leave some users 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 in equal size pieces as the optimal policy among the class of SSS polices, we establish a simple result comparing such policy with a non-sharing policy in which each stream of jobs are allocated exclusively to one particular server. This result are a special case of the performance analysis of a classical fork/join queueing problem ([2], [3], [4], [11], [15], and [16]) where the optimality of all equal sharing provides us with a rich structure we exploit. The extension of results in [2] and [3] (potentially much stronger than ours) to this instant remains to be a topic of future study.

Notation: From the definition of $\mathcal{A}_s$, we can simplify notations by replacing $A \in \mathcal{A}_s$ with its first column. For instance, we can write $D_\tau(\underline{\alpha}) := D_{\tau,i}(A(\underline{\alpha}))$, where $A(\underline{\alpha})$ is a cyclic matrix whole columns are permutations of $\underline{\alpha}$. Also, throughout the paper, 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. Furthermore, in Section V, we use the standard notion of $(\underline{\alpha}^i, \underline{\alpha}^{-i})$, from game theory, to describe policies of primary server i and others. In this case, assignment policy $A = (\underline{\alpha}^i, \underline{\alpha}^{-i})$ represents a matrix whose i^{th} column is vector $\underline{\alpha}^i$ (primary server i's splitting strategy) while $\underline{\alpha}^{-i}$ is an $n \times n-1$ represents the elements of the matrix except for the the i^{th} column.

III. Analysis of Problem P1

As mentioned before, in this Section, we restrict our attention to distribution matrices $A \in \mathcal{A}_s$. Theorm 1 below provides the first result of our paper.

Theorem 1: Assume G and λ are given and fixed. Among all $A = A(\alpha)$ such that $A \in \mathcal{A}_s$, the delay is minimized at $A^* = [1/n]_{n\times n}$, i.e.

$$E(D_i(1/n)) \leq E(D_i(\underline{\alpha})) \tag{3}$$

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

Proof: Remember that

$$E(D(\underline{\alpha})) = E\left(\max_j \{\alpha_j \tau_j + W_j(\underline{\alpha})\}\right) \tag{4}$$

$$E(D_\tau(1/n)) = E\left(\max_j \{\tau_j/n + W_j(1/n)\}\right) \tag{5}$$

Now we use the following Lemma.

Lemma 1: For any $\underline{\alpha}$, we have

$$W_i(1/n) \quad \leq_{st} \quad W_i(\underline{\alpha}) \tag{6}$$

where $\leq_{st}$ denotes the usual stochastic order defined on space of random vectors.

Now using this, we arrive at Lemma 2 below under the following assumption.

Assumption 1: The stationary queue backlogs $W_i(\underline{\alpha})$ and $W_j(\underline{\alpha})$ are independent.

Lemma 2: For any $\underline{\alpha}$, under Assumption 1, we have

$$E(\max_j \{\alpha_j \tau_j + W_j(1/n)\}) \leq E(\max_j \{\alpha_j \tau_j + W_j(\underline{\alpha})\}) \tag{7}$$

The proof of this lemma is given in the appendix. Furthermore, we use the following lemma to arrive at the assertion of the theorem.

Lemma 3: For any $\underline{\alpha}$, we have

$$E(\max_j \{\tau/n + W_j(1/n)\}) \leq E(\max_j \{\alpha_j \tau + W_j(1/n)\}) \tag{8}$$

The proof of this lemmas is also given in the appendix.

■

So far we have shown that among SSS policies, the policy which distribute the jobs equally among all servers is optimal. In the next section, we show that such policy outperforms the non-cooperative policy which allocates jobs of a stream exclusively to a particular server.

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 steam i to server i. The main result of this section is given by the following intuitive theorem:

Theorem 2: For a given load $\rho < 1$, the all-equal sharing policy outperforms the non-sharing policy.

Proof:

To start we 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 is nothing bu an M/G/1 system; we then use the Pollaczek-Khinchin formula to calculate the expected waiting time at each secondary queue. Using this we arrive at

$$E(W^{ns}) = \frac{\lambda E(\tau^2)}{2(1-\rho)}. \tag{9}$$

Similarly,

$$E(W_j(1/n)) = \frac{\lambda' E(\tau'^2)}{2(1-\rho')} \tag{10}$$

where we need to calculate arrival rate λ' and the first and second moments, $E(\tau')$ and $E(\tau'^2)$, associated with job size distributions at secondary queue i, when all equal sharing policy A^* is implemented. It is straight forward to verify that

$$\lambda' = n\lambda \tag{11}$$

$$E(\tau') = \frac{E(\tau)}{n} = \frac{1}{n\mu} \tag{12}$$

$$\rho' = \rho \tag{13}$$

$$E(\tau'^2) = \frac{E(\tau^2)}{n^2} \tag{14}$$

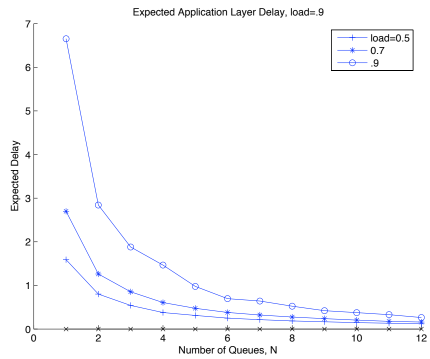

Fig. 2. Expected delay versus the number of servers to be shared

As a result, we have

$$E(W_j(1/n)) = \frac{\lambda E(\tau^2)}{2n(1-\rho)} = \frac{E(W^{ns})}{n} \tag{15}$$

With this we have

$$\begin{aligned} & E(D(1/n)) \\ = \; & E(\max_j(\tau_j/n + W_j(1/n))) \\ \leq \; & E(\max_j(\tau_j/n)) + E(\max_j(W_j(1/n))) \\ < \; & 1/\mu + E(\max_j(W_j(1/n))) \\ \leq \; & 1/\mu + nE(W_j(1/n)) \\ \leq \; & 1/\mu + \frac{\lambda E(\tau^2)}{2(1-\rho)} = E(D^{ns}) \end{aligned} \tag{16}$$

In other words, we have shown that for a given load $\rho < 1$, $E(D_\tau(1/n)) < E(D_\tau^{ns})$. From this we have the assertion of the theorem. ■

So far we have only presented existence result (in terms of parameters) for each system to outperform the other. Figures IV, shows the resulting average delays in simulations of M/M/1 queues versus number of (secondary) servers/queues to be shared. This result points out to the following generalization:

Conjecture 1: For any given a load $\rho < 1$, the all-equal sharing policy outperforms any equal sharing policy among a strict subset of servers, i.e.

$$E(\max_{1\leq j\leq n}(\frac{\tau_j}{n} + W_j(\frac{1}{n}))) \leq E(\max_{1\leq j\leq m}(\frac{\tau_j}{m} + W_j(\frac{1}{m}))) \tag{17}$$

This conjecture is based on the weak stochastic majorization and Schur convexity of the max function.

The most important consequence of the above theorem is to strengthen the optimality of policy A^* in the following sense: A^* not only outperforms all SSS policies, but also those equal-sharing policies that do not take advantage of all available servers.

In the next section, we show that when all users but one adhere to an all-equal sharing policy, no user can decrease her delay by unilaterally diviating from A^*.

V. Analysis of Problem P3

In this section, we show that the 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

$$D_i(\underline{\alpha}^{*i}, \underline{\alpha}^{*-j}) \leq D_i(\underline{\alpha}^i, \underline{\alpha}^{*-i}) \tag{18}$$

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

Proof:

- Recall that

$$E(D_i(\underline{\alpha}^i, \underline{\alpha}^{*-i})) = E(\max_j \{\alpha_j^i \tau_j + W_j(\underline{\alpha}, \underline{\alpha}^{*-i})\}) \tag{19}$$

$$E(D_i(\underline{\alpha}^{*i}, \underline{\alpha}^{*-i})) = E(\max_j \{\tau_j/n + W_j(1/n)\}) \tag{20}$$

Now we use the following lemma to arrive at the assertion of the theorem.

Lemma 4: Consider an arbitrary realization of the sequence of inter-arrival times and job sizes associated with all jobs arriving at primary queue i. Define $f(\underline{\alpha}^i) = E\left\{\max_j\left(\alpha_j^i \tau + W_j(\underline{\alpha}^i, \underline{\alpha}^{*-i})\right)\right\}$, where the expectation is taken with respect to the job sizes and arrivals to all other queues. Then $f(\underline{\alpha})$ is symmetric and convex in $\underline{\alpha}$.

From this lemma we have that for any given realization of the sequence of inter-arrival and job sizes of jobs into queue i, $E(D_i(\underline{\alpha}^i, \underline{\alpha}^{*-i}))$ is symmetric and convex in $\underline{\alpha}$. Now taking the expectation with respect to all jobs, we have that $E(D_i(\underline{\alpha}^i, \underline{\alpha}^{*-i}))$ is symmetric and convex in $\underline{\alpha}$. This means that we have

$$E(D_i(\underline{\alpha}^{*i}, \underline{\alpha}^{*-j})) \leq E(D_i(\underline{\alpha}^i, \underline{\alpha}^{*-i})), \tag{21}$$

hence, the assertion of the theorem. ■

Again proof of Lemma 4 is given in the appendix.

VI. Future Work

We have a long way in fully characterizing the delay improvement in a sharing system with parallel resources. 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. We hope to strengthen this result by taking advantage of analysis techniques applied to fork/join queuing problems.

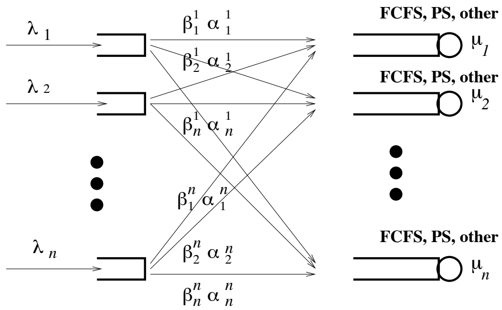

Fig. 3. A Generalization of the problem to include the impact of redundency

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 a system will mostly likely depend on the load as well as the degree of asymmetry among servers and arrival statistics. The ideas from stochastic ordering might not necessarily help us to provide optimal solutions, but will be helpful in providing comparisons between specific policies.

Furthermore, in the present paper we have ignored the issue of overhead/redundency. It is natural that sharing will add redundancy or an increase in the overhead. The analysis in the current paper is biased towards sharing policies as it ignores such impacts. The above generalizations are shown in Fig. VI.

In addition, we would like to extend the current result to other queueing disciplines. The results presented in this paper relies heavily on the basic idea of variance reduction (see [8]) to reduce waiting times in the first come first serve queues. Under different queue disciplines, statistical multiplexing might prove harder to capture. Extending our results to other queueing disciplines, such as generalized processor sharing or random access, is an interesting topic for future research.

On the technical side, also, there are interesting extensions to be considered. In establishing the optimality of A^* we relied on Assumption 1. Such an assumption may or may not be satisfied depending on application. While the server independency seem valid in case of wireless access point sharing, in the content sharing application it is expected to see a strong correlation among the upload time of pieces of a single file. In such cases, we believe the result of the current work still applies. The proof of this remains open.

In the absence of information about the state of queues, we have restricted our search for an optimal policies to those which can be described by stationary and fixed distribution matrices. It is important to note that in some applications (such as multi path routing) observation can provide the scheduler with imperfect information regarding the state of the secondary queues. Such information, even though not perfect, can be extremely valuable to the construction of an optimal policy and result in significant performance improvements. Similar notions of stochastic ordering and majorization are expected to arise in the space of probability measures (information state) as shown in [6], giving us great hope to extend the current work.

Acknowlegement

This research is in part funded by Center for Networked Systems research grant CNS0509. Furthermore, the author would like to thank Professor R. Cruz for many useful discussions and feedback.

References

[1] S. 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] F. Baccelli and Z. Liu. On the execution of parallel programs on multiprocessor systemsa queueing theoretic approach. *J. ACM*, 1990.

[3] F. Baccelli, A.M. Makowski, and A. Shwartz. The forkjoin queue and related systems with synchronization constraints: stochastic ordering and computable bounds. *Adv. Appl. Probab*, 1989.

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

[5] 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.

[6] F. Beutler and D. Teneketzis. Routing in queueing networks under imperfect information: Stochastic dominance and thresholds. *Stochastics and Stochastics Reports*, 1989.

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

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

[9] M. Harchol-Balter, M. Crovella, and C. Murta. On choosing a task assignment policy for a distributed server system. *Journal of Parallel and Distributed Computing*, November 1999.

[10] http://www.mushroomnetworks.com/v4/consumer/products.htm. Apx10 access point aggregator, for the enlightened wireless data user. Technical report.

[11] P. Konstantopoulos and J. Walrand. Stationarity and stability of forkjoin networks. *J. Appl. Probab*, 1989.

[12] 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.

[13] A. W. Marshall and I. Olkin. *Inequalities: Theory of Majorization and Its Applications*. Academic Press, 1979.

[14] A. Muller and D. Stoyan. *Comparison Methods for Stochastic Models and Risks*. Wiley, 2002.

[15] R. Nelson and A.N. Tantawi. Approximate analysis of fork/join synchronization in parallel queues. *IEEE Trans. Comput*, 1988.

[16] R. Nelson, D. Towsley, and A.N. Tantawi. Performance analysis of parallel processing systems. *IEEE Trans. Software Eng.*, 1988.

[17] W. Winston. Optimality of the shortest line discipline. *Journal of Applied Probability*, 14, 1977.

[18] CH. Xia, G. Michailidis, and N. Bambos. Dynamic on-line task scheduling on parallel processors. *Performance Evaluation*, 46(2-3), October 2001.

APPENDIX

In this appendix, we first prove Lemma 1.

Lemma 1: For any $\underline{\alpha}$, we have

$$W_i(1/n) \quad \leq_{st} \quad W_i(\underline{\alpha}) \tag{22}$$

where $\leq_{st}$ denotes the usual stochastic order defined on space of random vectors.

Proof: This lemma s a direct consequence of Theorems 6.3.5 of [14] applied to an M/G/1 system. All we need to notice is that a job arriving at secondary queue i can originate, with equal probability, at any of the primary queues, i.e. it has a random size $\tau_{\underline{\alpha}}$ where

$$\tau_{\underline{\alpha}} = \alpha_i \tau \qquad \text{with prob } \frac{1}{n}. \tag{23}$$

It is then easy to verify that $\tau_{\underline{\alpha}}$ stochastically dominates τ/n in a convex order sense (see [14]), i.e.

$$\tau/n \leq_{cx} \tau_{\underline{\alpha}}. \tag{24}$$

■

Before proceeding with proof of Lemma 2, we provide the following fact from [14] (Theorem 1.2.16):

Fact 1: If $\underline{X}$ and $\underline{Y}$ be vectors of independent random variable such that $X_i \leq_{st} Y_i$. For any function $g : \mathbb{R}^n \to \mathbb{R}^k$ that is increasing, $g(\underline{X}) \leq_{st} g(\underline{Y})$.

Lemma 2: For any $\underline{\alpha}$, we have

$$E(\max_j \{\alpha_j \tau + W_j(1/n)\}) \leq E(\max_j \{\alpha_j \tau + W_j(\underline{\alpha})\}) \tag{25}$$

Proof:

Consider an arbitrary realization of vector $(\tau_1, \ldots, \tau_n)$ of the service times associated with pieces of the job. We define $X_i = W_i(1/n)$ and $Y_i = W_i(\underline{\alpha})$. We now use Assumption 1, i.e. we assume that X_is and Y_is are independent. Also from Theorem 6.3.5 of [14] we have: $X_i \leq_{st} Y_i$. So now we use Fact 1 as follows. Given a fix vector of service times $(\tau_1, \ldots, \tau_n)$, function $g(\underline{x}) = \max_j \{\alpha_j \tau + x_j\}$ is an increasing function. From Fact 1 and the definition of stochastic order for finite mean variables, we have the theorem.

■

Before proceeding with the proof of Lemma 3, we revisit some useful definitions and establish Claims 1-2.

Definition 3: A set of n random variables are called *exchangeable* if the joint distribution of of the random vector is permutation invariant.

Note that two important special cases of exchangeable random variables are 1) when random variables are iid case and 2) when the random variables are identical (dependent).

Definition 4: Random variable Y is said to dominate random variable X in the *increasing convex order* sense, denoted by $X \leq_{icx} Y$, iff

$$\begin{aligned} &\iff \quad Ef(X) \leq Ef(Y) \quad \forall f \text{ increasing, convex.} \\ &\iff \quad \exists \hat{Y} =_{st} Y \text{ s.t. } E(\hat{Y}|X) \geq X \quad a.s. \end{aligned} \tag{26}$$

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

$$\overline{f}(y) = \frac{1}{n} \sum_{i=1}^{n} f_i(y). \tag{27}$$

Then

$$\max_i \overline{f}(X_i) \leq_{icx} \max_i f_i(X_i). \tag{28}$$

Proof: To prove the claim, we construct random variable $Y = \max_i f_{\pi(i)}$ where π 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:

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

■

Similarly we can simply extend this to a bivariate case:

Claim 2: Let $X_1, \ldots, X_n$ and $T_1, \ldots, T_n$ be two independent set of exchangeable random variables and $f_1, \ldots, f_n$ measurable real functions. Define the function $\overline{f}$ by

$$\overline{f}(y, t) = \frac{1}{n} \sum_{i=1}^{n} f_i(y, t). \tag{29}$$

Then

$$\max_i \overline{f}(X_i, T_i) \leq_{icx} \max_i f_i(X_i, T_i). \tag{30}$$

Lemma 3: For any $\underline{\alpha}$, we have

$$E(\max_j \{\tau_j/n + W_j(1/n)\}) \leq E(\max_j \{\alpha_j \tau_j + W_j(1/n)\}) \tag{31}$$

Proof:

Noting the exchangeability of $W_j(1/n)$ and τ_j, we put $f_j(x,t) = \alpha_j t + x$ and $\overline{f}(x) = t/n + x$. Using the first line of (26), we arrive at the assertion of the lemma. ■

To prove Lemma 4, we use the following fact (Proposition B.4 in [13]):

Fact 2: Let $X_1, \ldots, X_n$ be exchangeable random variables. Let $\Phi(\underline{x}; \underline{\alpha}) = \phi(w(x_1, a_1), \ldots, w(x_n, a_n))$ 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,

$$\psi(\underline{\alpha}) = E\Phi(\underline{X}; \underline{\alpha}) \tag{32}$$

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

Lemma 4: Consider an arbitrary realization of the sequence of inter-arrival times and job sizes associated with jobs arriving at primary queue i. Define $f(\underline{\alpha}^i) = E\left\{\max_j \left(\alpha_j^i \tau + W_j(\underline{\alpha}^i, \underline{\alpha}^{*-i})\right)\right\}$, where the expectation is taken with respect to the job sizes and arrivals to all other queues. Then $f(\underline{\alpha})$ is symmetric and convex in $\underline{\alpha}$.

Proof: Without loss of generality, put $i = 1$.Furthermore, without loss of generality, consider an arbitrary realization of the sequence of inter-arrival times and job sizes associated with the jobs arriving at primary queue 1.

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

- τ_{-1}: These are pieces of those jobs that have arrived at any primary queues but primary queue 1. These jobs have been split equally and sent to all n secondary queues (all primary servers but server 1 follows the all equal sharing); their arrival times and sizes are identical across queues.
- $\tau_{1,j}$: These are pieces of those jobs arriving at primary queue 1 which have been forwarded to secondary queue j. Their distribution depends on α_j^1 and their realization is assumed fixed in this lemma.

Now notice that across all secondary queues, $\{\tau_{-1}\}$ and their corresponding inter-arrival times $\{B_{-1}\}$ 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 the jobs of primary queue 1, hence, a function of α_j. In other words, we can put $\alpha_j^1 \tau + W_j(\underline{\alpha}^1, \underline{\alpha}^{*-1}) = w(\{X\}, \alpha_j)$, where $\{X\}$ is a short hand for the vector of random variables associated with all basic random events in the system, i.e. $\{\tau_{-1}\}$ and $\{B_{-1}\}$ except for those of the large jobs arrived at primary queue 1. Note that for given realization of $\{\tau_{-1}\}$ and $\{B_{-1}\}$, w is a convex in α_j (see section 6.3 in [14]). On the other hand, notice that $\phi(\underline{y}) = \max(\underline{y})$ is convex, symmetric and increasing. Now taking expectation sequentially on each variable of sequence $\{X\}$ and using Fact 2, we arrive at the assertion of the lemma.

■