Inducing Optimal Service Capacities via Performance-Based Allocation of Demand in a Queueing System with Multiple Servers

Sin-Man Choi, Ximin Huang and Wai-Ki Ching
Advanced Modeling and Applied Computing Laboratory, Department of Mathematics,
The University of Hong Kong, Pokfulam Road, Hong Kong.
E-mail: kellyci@hkusua.hku.hk, hehe1121@hkusua.hku.hk, wching@hkusua.hku.hk.

***Abstract*—In this paper, we study the use of performance-based allocation of demand in a multiple-server queueing system. The same problem with two servers have been studied in the literature. Specifically, it has been proposed and proved that the linear allocation and mixed threshold allocation policies are, respectively, the optimal state-independent and state-dependent allocation policy in the two-server case. The multiple-server linear allocation has also been shown to be the optimal state-independent policy with multiple servers. In our study, we focus on the use of a multiple-server mixed threshold allocation policy to replicate the demand allocation of a given state-independent policy to achieve a symmetric equilibrium with lower expected sojourn time. Our results indicate that, for any given multiple-server state-independent policy that prohibits server overloading, there exists a multiple-server mixed threshold policy that gives the same demand allocation and thus have the same Nash equilibrium (if any). Moreover, such a policy can be designed so that the expected sojourn time at a symmetric equilibrium is minimized. Therefore, our results concur with previous two-server results and affirm that a trade-off between incentives and efficiency need not exist in the case of multiple servers.**

***Index Terms*—queueing system, threshold allocation, game theory**

I. Introduction

The problem of finding the optimal control policy for a queueing system has been widely studied in the literature [1]–[3], [5], [6]. Recent studies have focussed on queueing system with strategic servers [2], [6], particularly on deriving an optimal policy to induce high service capacities in a competitive environment [1], [3], [5]. In these systems, the servers decide their own service capacities and compete with each other. It is then of interest what kind of policy for customer allocation and compensation can be used to induce high service capacities from the servers with minimum cost.

Among different means to motivate faster service, the use of demand allocation to achieve this goal has first been studied by Gilbert and Weng in [5], where a common-queue allocation policy and a separate-queue allocation policy were compared in a two-server setting. Their results have been extended in [3] to the case of multiple servers. On the other hand, Cachon and Zhang [1] have further explored the two-server problem where the buyer can use demand allocation policy which explicitly accounts for the servers' chosen service capacities. A wider range of allocation policies were then studied, and it was concluded that the optimal policies in both classes can induce the maximum feasible service capacity, and thus there are no trade-off between incentives and efficiency.

The study in Cachon and Zhang [1] was based on a two-server queueing system where both service times and inter-arrival times are exponentially distributed. In the demand allocation problem, the buyer would like to induce a high service capacity from strategic servers through a performance-based allocation of demand and a compensation proportional to allocated demand. Two classes of allocation policies, namely the state-independent allocation policies and the state-dependent allocation policies were studied and compared. They showed that linear allocation policy is an optimal state-independent policy and induces the maximum feasible service capacity from servers. The same result for the case of more than two servers can be found in [10]. They further argued that by randomizing between two-server threshold allocation policies, one could achieve allocation identical to the linear allocation policy. They also claimed that an optimal state-dependent policy exists. We remark that in cases where the capacity of the primary server is lower than the total demand rate, this is only possible if we allow ourselves to allocate customers only to the primary server and pay the server at allocation, which makes the system unstable even when the total service capacity is greater than the total demand rate.

The main aim of this paper is to extend the mixed threshold policies proposed in [1] to multiple-server mixed threshold policies, and study to what extent they can replicate the demand allocation of state-independent policies. Our result shows that, if we restrict the compensation for each customer to be paid at the time of service completion and prohibits overloading a server, then the multiple-server mixed threshold policies can replicate the demand allocation of any state-independent policy. The replication of the demand allocation of a state-independent policy with server overloading and payment at customer allocation is feasible if we allow the inclusion of single-sourcing (with payment at customer allocation) with some probability in the mixed threshold policy. Moreover, assuming that all servers are identical, in the Nash equilibrium, the expected sojourn time with our mixed threshold policy is optimal with the equilibrium service capacities. In other words, our results concur with the two-server results of [1] and indicate that there is no tradeoff between incentive and efficiency.

The rest of the paper is structured as follows. Section II introduces the multiple-server demand allocation problem and review previous results obtained by [1] and [10]. In Section III, we generalize the two-server threshold policy to an n-server threshold policy and show the set of demand allocation that can be replicated using an n-server mixed threshold policy. In Section IV, we summarize the results and discuss further research issues.

II. The Multiple-server Demand Allocation Problem

We consider a queueing system with n identical strategic servers. Each server decides its own service capacity μ_i and incurs a cost at the rate of $c(\mu_i)$, where $c(.)$ is assumed to be strictly increasing and convex, i.e. $c'(.) > 0$ and $c''(.) \geq 0$. The time that Server i serves a customer is, independent of all other service times, exponentially distributed with mean rate μ_i. Customers arrive at the system according to a Poisson process with rate λ. The buyer pays each server an amount of R for each customer it completes serving. The aim of the buyer is to select an demand allocation policy, through which the customers are assigned to the servers, that minimizes the expected sojourn time for a customer in the equilibrium.

For the expected waiting times to be finite in an equilibrium where the n servers split the demand equally, we require

$$c(\frac{\lambda}{n}) < \frac{\lambda R}{n}. \tag{1}$$

Moreover, as a benchmark for comparison, we define the *maximum feasible service capacity* as $\bar{\mu}$ where $c(\bar{\mu}) = \frac{\lambda R}{n}$. In other words, the maximum feasible service capacity is the service capacity at which, when chosen by all servers, each server receives equal share of the demand and earns zero profit.

A. State-independent and State-dependent Allocation Policies

As proposed in [1], there are two classes of allocation policies, namely *state-independent* and *state-dependent* allocation policies. Under state-independent policies, customer allocation is only based on the service capacities of the servers, but not the states of the servers (i.e., whether it is busy or idle). Customers can be immediately allocated to a server upon arrival and a first-in-first-out queue is maintained for each server. We further assume that the arrival of customers to each of the servers follows a Poisson process with rate λ_i, which can be achieved by allocating each customer to Server i with probability λ_i/λ. Examples of state-independent policies with multiple servers are the separate-queue allocation [3], [5], linear allocation and the proportional allocation [1], [10]. In particular, in [10] it was proved that the n-server linear allocation policy is optimal given that we pay the servers for the customers at allocation.

The other class of allocation policies, the state-dependent policies, allows customer allocation to depend on the current state of the servers. The most common example is the common-queue allocation policy [3], [5], but we will focus on the n-server extension of the mixed threshold policy discussed in [1].

B. State-independent Policies: A Review of the Multiple-server Linear Allocation Policy

The two-server linear allocation policy proposed by Cachon and Zhang [1] has been shown to be an optimal state-independent policy with appropriate parameters chosen. The policy and results in the case of n servers have been studied by Zhang in [10]. Under the multiple-server linear allocation policy, the allocation to Server i is given by

$$\lambda_i(\mu) = \begin{cases} \theta\mu_i^\rho - \frac{1}{\hat{n}}\left(\theta\sum_{j=1}^{\hat{n}}\mu_j^\rho - \lambda\right) & i \leq \hat{n}. \\ 0 & i > \hat{n}. \end{cases} \tag{2}$$

Here the servers' capacities are sorted in a decreasing order, $\theta > 0$, $0 < \rho \leq 1$ and $\hat{n} \leq n$ is the largest integer such that $\lambda_{\hat{n}} \geq 0$ and $\mu_{\hat{n}} > 0$.

It should be noted that under this n-server linear allocation, the demand allocated to Server i can be greater than the service capacity chosen by Server i, i.e., $\lambda_i > \mu_i$ with some capacity vector $\boldsymbol{\mu}$. Moreover, for mathematical tractability and simplicity, [10] has adopted the assumption that the servers are paid for the job at allocation instead of completion. In other words, there are cases where a server is paid for more customers than it can actually serve, but such cases do not occur in the Nash equilibrium of the game.

Under the assumption that the servers are paid for the job at allocation, Zhang [10] has proved the existence and uniqueness of a Nash equilibrium in which the service capacity equals to the maximum feasible service capacity when the appropriate values of θ and ρ are chosen. Specifically, when the cost function $c(\mu_i)$ is strictly convex, Zhang [10] proved that a unique equilibrium exists with

$$\theta = \frac{nc'(\mu_l)}{R(n-1)} \tag{3}$$

and $\rho = 1$ when $R > r_1 = c(\lambda/n)/(\lambda/n)$. In the equilibrium $\mu_i = \mu_l = \bar{\mu}$ for all i and expected service times are finite. For the case where the cost function $c(\mu_i)$ is linear, i.e., $c(\mu_i) = b\mu_i \quad (b > 0)$, Zhang [10] proved that a unique equilibrium exists with

$$\theta = \frac{n}{n-1}\left(\frac{2b\mu_l^{1/2}}{R}\right) \tag{4}$$

and $\rho = 1/2$ when $R > r_1 = c(\lambda/n)/(\lambda/n)$. In the equilibrium $\mu_i = \mu_l = \bar{\mu}$ for all i and the expected service times are finite.

C. State-dependent Policies: A Review of the Two-server Mixed Threshold Allocation Policy

The two-server threshold allocation has been studied as a control policy with non-strategic servers in the literature. In particular, it has been proved in [7] that a buyer's optimal allocation with two non-strategic servers is of threshold type. Under a two-server threshold allocation, a single queue is maintained for the two servers, but a job may not be allocated immediately to a server upon arrival, even if the server is idle. Job allocation is based on the designation of the primary (and secondary) server and a threshold parameter m. When a job arrives, it is allocated to the primary server if it is idle or has fewer than m jobs in queue and allocated to the secondary server only if it is idle, the primary server is busy, and has m jobs in queue. A numerical method for evaluating the system's performance under threshold allocation has been studied by Rubinovitch [9]. It can be seen that, when different values of m are chosen, the demand allocation to the servers would be different.

In [1], Cachon and Zhang studied the two-server allocation problem with strategic servers. They proposed randomizing between the threshold allocation with different parameters to replicate the demand allocation of the linear allocation policy, so that the maximum feasible service capacity can be attained in the Nash equilibrium. Specifically, they argued that the buyer can allocate any portion of the buyer's demand to the primary server by varying which server is designated the primary server and randomizing among different threshold values m.

However, it is worth noting that, when the primary server's service capacity μ_1 is less than the total demand λ, with any finite values of m, the secondary server is allocated at least $\lambda - \mu_1$ of demand. The limit of the primary server's demand, as m goes to infinity, is μ_1. The only way to allocate more than μ_1 demand to the primary server is not to use the secondary server at all, i.e. setting $m = \infty$ and making $\lambda_i = \lambda$, and to pay for the customers to the server at allocation instead of service completion. This will cause the system to be unstable, even in cases where $\mu_1 + \mu_2 > \lambda$, and is undesirable. It is therefore important to identify the state-indepedent policies which can be replicated by a mixed threshold policy without overloading (i.e. a policy such that $\lambda_i \leq \mu_i$).

III. Multiple-Server Threshold Policies

In this section, we will generalize the two-server threshold policy to an n-server threshold policy. We will follow the convention that the buyer pays the server for a customer when the service is completed.

A. The n-server Policy

With n servers, where $n \geq 2$, it is natural to extend the two-server case by assigning the servers as the $1^{st}, 2^{nd}, \ldots, n^{th}$ servers and specifying $n-1$ threshold parameters. Similar control policies for non-strategic servers have been studied in [8]. In some of these studies the threshold parameters may depend on the state of the other servers. (More precisely, the threshold for the i^{th} server can be different depending on the state of the $(i+1)^{th}, \ldots, n^{th}$ servers). However, for simplicity and due to the reason that randomization gives enough flexibility for parameterizing the policy, we shall assume that m_i is a constant for each policy in our study.

A n-server (pure) threshold allocation policy T is specified by an assignment of the Servers $1, 2, \ldots, n$ as the $1^{st}, 2^{nd}, \ldots, n^{th}$ servers and the thresholds $m_2, \ldots, m_n$ where each m_i is a nonnegative integer. We define $m_1 = 0$. A single queue is maintained in the system. When a customer arrives, it is assigned to Server 1 if it is idle. If Servers $1, 2, \ldots, i-1$ are all busy and the number of waiting customers (including the new arrival) is more than $m_1 + \ldots + m_i$, the customer is assigned to Server i. Otherwise, it waits in the queue. When a Server i completes service of a customer, if the number of waiting customers is more than $m_1 + \ldots + m_i$, then the first customer in the queue is assigned to Server i. If $m_i = \infty$ for some i, then no customer is allocated to Servers $i, i+1, \ldots, n$.

For any service capacity vector $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_n)$, the demand allocated to the servers via the allocation policy T is $\boldsymbol{\lambda}^{(T)}$, where $\lambda_i^{(T)}$ is defined to be Server i's expected rate of receiving customers. In each state, if Server i is idle, its rate of receiving customers is the arrival rate of customers multiplied by the probability that an arriving customer is allocated to Server i. On the other hand, if Server i is busy and there are customers waiting in the system, its rate of receiving customers is μ_i if the waiting customers can be assigned to Server i upon service completion of the current customer, and is zero otherwise. Because the n-server policy is much more complicated, it is not straightforward to see what demand allocation can be achieved by the pure policy and by randomizing between some n-server threshold policies. In the following, we give some properties of an n-server pure threshold allocation policy.

Lemma 1: Given n servers with service capacity vector $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)$ where $\sum_{i=0}^{n} \mu_i > \lambda$. Suppose Server i is designated as the i-th server. Let $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n)$ be the allocated demand of an n-server pure threhold policy with threshold $m_2, m_3, \ldots, m_n$. We have the following:

(i) Let $k = \max\{i : 1 \leq i \leq n, m_j < \infty \ \ \forall j = 1, 2, \ldots, i\}$. Then the system is stable if and only if $\sum_{i=1}^{k} \mu_i > \lambda$. When the system is stable, we have $\lambda = \sum_{i=1}^{n} \lambda_i$.

(ii) If $m_i = \infty$, then $\lambda_j = 0$ for all $j = i, i+1, \ldots, n$.

(iii) For any $i = 2, \ldots, n$, given fixed and finite values of m_j for $j = 1, \ldots, i-1$, then for any $\epsilon > 0$, there exists m_i^* such that for any $m_{i+1}, \ldots, m_n$ and $m_i > m_i^*$ we have

$$\min\left(\sum_{j=1}^{i-1} \mu_j, \lambda\right) \geq \sum_{j=1}^{i-1} \lambda_j > \min\left(\sum_{j=1}^{i-1} \mu_j, \lambda\right) - \epsilon.$$

All proofs can be found in [4].

We have seen that the demand allocation to the servers can be varied by adjusting the thresholds of a policy. However, because the thresholds only take integral values, the demand allocation is limited to a countable set of points. To enable us to select from a wider range of demand allocation, we introduce the n-server mixed policy, which randomizes between a number of pure threshold policies.

Definition 1: An n-server mixed threshold allocation policy τ is specified by an integer $k \geq 1$, real numbers $\alpha_1, \ldots, \alpha_k$ such that $\sum_{i=1}^{k} \alpha_i = 1$ and k n-server threshold policies $T_1, \ldots, T_k$. When the mixed threshold allocation policy is used, each of the threshold policy T_i is used with probability α_i. The demand allocated via the mixed threshold policy τ is then denoted by $\boldsymbol{\lambda}^{(\tau)}$ and given by

$$\lambda_j^{\tau} = \sum_{i=1}^{k} \alpha_i \lambda_j^{T_i}$$

for any server $j = 1, 2, \ldots, n$.

It is clear that the set of demand vectors that can be allocated by a pure threshold policy when $\sum_{i=1}^{n} \mu_i > \lambda$ is contained in the set

$$S_{\boldsymbol{\mu}} = \{\boldsymbol{\lambda^t} : 0 \leq \lambda_i^t \leq \min(\mu_i, \lambda) \text{ and } \sum_{i=1}^{n} \lambda_i^t = \lambda\}. \quad (5)$$

Since $S_{\boldsymbol{\mu}}$ is a convex set, it follows immediately that the set of feasible allocation vectors, i.e., the set of demand vectors that can be allocated by a mixed threshold policy, is also contained in $S_{\boldsymbol{\mu}}$. In the following, we explore which allocation vectors in $S_{\boldsymbol{\mu}}$ can be achieved by mixed threshold policy given a fixed service capacity vector $\boldsymbol{\mu}$ such that $\mu_1 + \ldots + \mu_n > \lambda$. Unless otherwise specified, we shall assume that we have such a fixed service capacity vector in the following.

Suppose we have $\boldsymbol{\lambda^t}$ such that $\lambda_1^t + \ldots + \lambda_n^t = \lambda$. We say that an allocation policy τ with demand allocation $\boldsymbol{\lambda}$ is *$\boldsymbol{\lambda}^t$-dominated in the order* $(i_1, i_2, \ldots, i_n)$ if

$$\sum_{j=l}^{n} \lambda_{i_j} \leq \sum_{j=l}^{n} \lambda_{i_j}^t \quad \text{for all } l = 2, \ldots, n. \quad (6)$$

where $i_1, i_2, \ldots, i_n \in \{1, 2, \ldots, n\}$ and are distinct.

We note also that the above condition implies that $\lambda_{i_1} \geq \lambda_{i_1}^t$ since

$$\sum_{j=1}^{n} \lambda_{i_j} = \sum_{j=1}^{n} \lambda_{i_j}^t = \lambda. \quad (7)$$

Lemma 2: Given n servers with service capacity vector $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)$ such that $\mu_1 + \mu_2 + \ldots + \mu_n > \lambda$, and an allocation vector $\boldsymbol{\lambda^t} = (\lambda_1^t, \lambda_2^t, \ldots, \lambda_n^t)$ such that $\lambda_1^t + \lambda_2^t + \ldots + \lambda_n^t = \lambda$. If $\lambda_j^t < \min(\mu_j, \lambda)$ for all $j = 1, 2, \ldots, n$, then there exists an n-server (pure) threshold policy that is $\boldsymbol{\lambda}^t$-dominated in the order $(1, 2, \ldots, n)$.

The pure threshold policies in Lemma 2 will be used in the following to compose a mixed threshold policies that gives the our target demand allocation. The following two lemmas are used to show that we can construct mixed threshold policies with some nice properties for the construction of the one that fulfills our goal.

To facilitate our discussion, we say that an allocation policy τ with demand allocation $\boldsymbol{\lambda}$ is *$\boldsymbol{\lambda}^t$-dominated and m-smaller in the order* $(i_1, i_2, \ldots, i_n)$ if the policy is $\boldsymbol{\lambda}^t$-dominated in the order $(i_1, i_2, \ldots, i_n)$ and $\lambda_{i_j} \leq \lambda_{i_j}^t$ for all $j = m, m+1, \ldots, n$, where m is an integer such that $2 \leq m \leq n$ and $i_1, i_2, \ldots, i_n \in \{1, 2, \ldots, n\}$ and are distinct.

Note that in the above definition, the property is equivalent up to any permutation of $i_m, i_{m+1}, \ldots, i_n$. Also, note that any policy $\boldsymbol{\lambda}^t$-dominated in the order $(i_1, i_2, \ldots, i_n)$ is $\boldsymbol{\lambda}^t$-dominated and n-smaller in the order $(i_1, i_2, \ldots, i_n)$.

Lemma 3: For fixed $\boldsymbol{\mu}$ and fixed $m \in \{3, \ldots, n\}$, suppose for each $k = m, m+1, \ldots, n-1$, we have a mixed threshold policy $\tau_{m,k}$ that is $\boldsymbol{\lambda}^t$-dominated and m-smaller in the order $(1, 2, \ldots, m-2, k, m-1, m, \ldots, k-1, k+1, \ldots, n)$. Then there exists a mixed threshold policy τ_{m-1} that is $\boldsymbol{\lambda}^t$-dominated and $(m-1)$-smaller in the order $(1, 2, \ldots, n)$.

Lemma 4: Given n servers with service capacity vector $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)$ such that $\mu_1 + \mu_2 + \ldots + \mu_n > \lambda$, and an allocation vector $\boldsymbol{\lambda^t} = (\lambda_1^t, \lambda_2^t, \ldots, \lambda_n^t)$ such that $\lambda_1^t + \lambda_2^t + \ldots + \lambda_n^t = \lambda$. If $\lambda_j^t < \min(\mu_j, \lambda)$ for all $j = 1, 2, \ldots, n$, then for any fixed k, there exists an n-server mixed threshold policy such that $\lambda_k \geq \lambda_k^t$ and $\lambda_j \leq \lambda_j^t$ for all $j \neq k$.

Proposition 1: Given some fixed service capacity vector $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)$ and target allocation vector $\boldsymbol{\lambda^t} = (\lambda_1^t, \ldots, \lambda_n^t)$ with $\sum_{i=1}^{n} \lambda_i^t = \lambda$ and $0 < \lambda_i^t < \min(\mu_i, \lambda)$ for $i = 1, 2, \ldots, n$. Then there exists a mixed threshold allocation policy such that $\lambda_i = \lambda_i^t$ for all $i = 1, 2, \ldots, n$.

The above lemmas show that for any $\boldsymbol{\mu}$ with $\mu_1 + \mu_2 + \ldots + \mu_n > \lambda$, any demand allocation vector set in the interior of the set $S_{\boldsymbol{\mu}}$ is the allocated demand of some mixed threshold policy. Moreover, if $\lambda_i^t = 0$ for some i, the allocated demand can be achieved by removing all servers with $\lambda_i^t = 0$ and considering a mixed threshold policy for the reduced set of servers. On the other hand, if $\lambda_i^t = \lambda$ for some i, then it can be achieved by only using Server i. Therefore, the set

$$S'_{\boldsymbol{\mu}} = \{\boldsymbol{\lambda^t} : 0 \leq \lambda_i^t < \mu_i, \lambda_i^t \leq \lambda \text{ and } \sum_{i=1}^{n} \lambda_i^t = \lambda\} \quad (8)$$

is achievable. It remains to investigate whether we could find a mixed threshold policy that achieves $\boldsymbol{\lambda^t}$ when $\lambda_i^t = \mu_i < \lambda$ for some i. However, this is impossible, because whenever other servers are used, the demand allocated to Server i, λ_i, would be strictly less than μ_i. In order to have λ_i to equal μ_i, we must not use any of the other servers. The remaining demand $\lambda - \mu_i$ then cannot be allocated to other servers and the system would be unstable. Thus it is impossible to allocate to a Server i exactly a demand of μ_i using a threshold allocation if all demand has to be allocated.

Still, there are two ways to solve the problem. First, since λ_i approaches μ_i in the limit, for any small number $\epsilon > 0$ one can find a value of the threshold such that $|\mu_i - \lambda_i| < \epsilon$. Second, one can use a state-independent allocation and assign a proportion of μ_i/λ of the arrivals to Server i for such cases.

B. Analysis on Unstable Queueing System

In the above sections, we have mainly focussed on the case where the total service capacities exceed the total demand rate and so all demand are allocated, i.e. $\sum_{i=1}^{n} \lambda_i = \lambda$. If the sum of the chosen service capacities are less than the total demand rate, $\mu_1 + \ldots + \mu_n \leq \lambda$, the queueing system is not stable regardless of the values of $m_2, \ldots, m_n$. Although it is natural to utilize the servers as much as possible when the system is not stable, the alternative may be useful with strategic servers to induce the servers to switch to higher service capacities in the long run, since we are mainly concerned with the equilibrium service capacities. Technically, designing an allocation policy that assigns $\lambda_i < \mu_i$ to Server i in these cases may help to prevent the existence of an undesirable Nash equilibrium where the queueing system is unstable.

In [1], under the state-independent linear allocation, a server may be given an allocated demand more than, equal to or less than its service capacity when the queueing system is not stable. We remark that with threshold allocation, when the system is unstable, it remains impossible to allocate to a server a demand level that is higher than its capacity, because a customer is only assigned to the server when it is idle. Thus any demand allocation where $\lambda_i > \mu_i$ is not possible. As a pure strategy, the buyer can choose to allocate a demand of zero or μ to Server i by setting m_i to be infinite or finite, respectively. Under the condition that $\mu_1+\mu_2+\ldots+\mu_n \le \lambda$, the threshold m_i does not affect the allocated demand of other servers. Consequently, we can randomize between the values of m_i and obtain any allocated demand λ_i such that $0 \le \lambda_i \le \mu_i$. Therefore we conclude that the set of feasible allocation when $\mu_1+\mu_2+\ldots+\mu_n < \lambda$ is the set of allocation vectors satisfying $0 \le \lambda_i \le \mu_i$.

C. Efficient Mixed Threshold Policies

We have shown that the set of demand allocation vectors

$$S_{\boldsymbol{\mu}} = \begin{cases} \{\boldsymbol{\lambda^t} : 0 \le \lambda_i^t \le \min(\mu_i, \lambda) \text{ and } \sum_{i=1}^n \lambda_i^t = \lambda\} \\ \quad \text{for} \quad \sum_{i=1}^n \mu_i > \lambda \\ \{\boldsymbol{\lambda^t} : 0 \le \lambda_i^t \le \min(\mu_i, \lambda)\} \quad \text{for} \quad \sum_{i=1}^n \mu_i \le \lambda \end{cases} \tag{9}$$

can be replicated by mixed threshold policies. However, it is not yet certain whether such policies perform better than the state-independent policies. It has been shown that for servers with different service capacities, the optimal policy that gives the lowest expected sojourn time is of threshold type [8], but some thresholds may depend on the states of the other servers, and the mixed threshold policy we discussed earlier may not give the lowest expected sojourn time with respect to the chosen service capacities. Indeed, in order to design an allocation policy that induces the server to choose the maximum feasible capacity, efficiency must be given up with some out-of-equilibrium choices of service capacities. Hence, our aim in this section is to find out whether the mixed threshold policy can give a lower expected sojourn time *in equilibrium* when compared to the state-independent policies.

As we deal with identical servers, we expect a symmetric equilibrium, where all servers choose the same service capacity and receive equal share of the demand. It is desirable that our mixed threshold policy gives the minimal expected sojourn time in this case, which will be shown in the following two propositions:

Proposition 2: When $\mu_1 = \mu_2 = \ldots = \mu_n = \mu_c > \lambda/n$, we can randomize among some threshold allocation policies with zero thresholds to obtain the allocation $\lambda_1 = \lambda_2 = \ldots = \lambda_n = \lambda/n$.

Proposition 3: In an n-server queueing system, given that $\mu_1 = \mu_2 = \ldots = \mu_n = \mu_c$, any n-server threshold allocation with all thresholds being zero gives the same expected sojourn time as an n-server common-queue system where each server has service capacity μ_c.

Finally, note that because any pure threshold policy with all threshold being zeros has an expected sojourn time identical to that of the n-server common queue, any mixed policy that is composed of such pure threshold policies would have the same expected sojourn time too. Combining with Proposition 2, we have shown that the mixed threshold policy used to replicate a state-independent policy could be designed to have minimal sojourn times in a symmetric equilibrium that is better than the state-independent policy. Thus the use of a mixed threshold policy could indeed help to improve efficiency and lower the expected sojourn time in the equilibrium.

D. Interpretations and Discussions

We have shown that for any fixed service capacity vector $\boldsymbol{\mu}$ and any target demand allocation vector $\boldsymbol{\lambda}$ such that $0 \le \lambda_i \le \lambda$, $\lambda_i < \mu_i$ and $\lambda_1+\ldots+\lambda_n = \lambda$ (if $\mu_1+\ldots+\mu_n > \lambda$), it is possible to choose a mixed threshold policy that gives the demand allocation $\boldsymbol{\lambda}$. The case where $\lambda_i = \mu_i$ can be catered for by using a state-independent allocation. Applying the respective policy for each service capacity vector when it is observed, we have a state-dependent policy that gives the demand allocation $\boldsymbol{\lambda(\mu)}$. In other words, for any state-independent policy P_1 with demand allocation $\boldsymbol{\lambda}$ such that $0 \le \lambda_i \le \min(\mu_i, \lambda)$, there exists a state-dependent policy that replicates the demand allocation of the policy P_1. Moreover, from the discussion in section 3.3, we see that the expected sojourn time under the state-dependent policy is lower than that under policy P_1. We conclude that for any state-independent policy that does not overload the servers, i.e., $\lambda_i \le \mu_i$, there exists a state-dependent policy that replicates the same demand allocation, thus giving the same Nash equilibrium but a lower expected sojourn time in the equilibrium.

We then discuss the case where server overloading is permitted. In the above, we have always imposed the conditions $\lambda_i \le \mu_i$ for $i = 1, 2, \ldots, n$ and assumed that the servers are paid at service completion. However, as we have seen in [10], there could be a state-independent allocation that gives an optimal equilibrium but does not satisfy the above criteria. To replicate the allocation of such policies, we must allow servers to be overloaded and paid at customer allocation instead of service completion.

If we assign all the demand to one server, say Server i, and pay the server at customer allocation, then the demand allocated to Server i and its rate of revenue, would be λ and λR respectively. Randomizing this allocation with other mixed threshold policies, it is possible to achieve any target demand allocation $\boldsymbol{\lambda}$ such that $0 \le \lambda_i \le \lambda$ and $\sum_{i=1}^n \lambda_i = \lambda$. This can be easily proved by noting that allocating all demand to Server i gives the demand allocation $\boldsymbol{\lambda} = \mathbf{e}_i =$

$(0,\ldots,0,\underbrace{1}_{i^{th}\text{entry}},0,\ldots,0)$, for $i = 1,\ldots,n$, and any target demand allocation can be expressed as a convex combinations of these vectors. However, such an allocation results in infinite waiting times and should be avoided as far as possible. Thus, for demand vectors such that $0 \leq \lambda_i \leq \min(\mu_i, \lambda)$, we can apply the results in previous subsections and use a mixed threshold policy that comprises of only threshold policies with finite threshold to replicate the demand allocation. In particular, at equilibrium we only need to randomize between threshold policies with zero thresholds, so that the expected sojourn time is equal to that in an n-server common queue system with the maximum feasible service capacity chosen.

IV. Concluding Remarks

In this paper, we have extended the two-server mixed threshold allocation policy proposed by by [1] to the case of n servers. For any state-independent policy that prohibits server overloading, we have shown that it is possible replicate the demand allocation by a mixed threshold allocation policy. For state-independent policy that includes server overloading, we have also shown that a mixed threshold allocation policy can replicate the allocated demand if we include a single-sourcing strategy in the mixed policy and allows payment at customer allocation. For identical servers, the mixed threshold policy at the symmetric equilibrium can be composed of only threshold policies with zero thresholds. As a result, it provides the minimal expected sojourn time with the equilibrium service capacities.

Our results concur with existing results with two servers that there are no trade-off between incentive and efficiency. Whether or not we allow server overloading, we can find a n-server mixed threshold policy that induces the same service capacity from the servers as any given state-independent policy. Moreover, in the symmetric equilibrium, the mixed threshold policy always gives a lower expected sojourn time.

Our results have been derived under several assumptions. First, we have assumed that all servers are identical, i.e. they have the same cost function $c(\mu)$. Nevertheless, the results in Section III-A and III-B are independent of the cost structure of the servers. Therefore, with asymmetric servers, it is also possible to replicate the demand allocation of any state-independent policy that prohibits server overloading by an n-server threshold allocation policy. However, because the Nash equilibrium, when exists, may not be symmetric, it has yet to be investigated whether a suitable n-server threshold allocation policy gives a lower expected sojourn time than a state-independent policy. Second, our model is based on a Markovian queueing system. A similar analysis can be carried out in the cases with more general distributions of the inter-arrival times or service times. However, the computation of the n-server mixed threshold policy may be more difficult due to the difficulty in the computation of the allocated demand in a n-server threshold system. Thirdly, we have assumed that the service capacities chosen by the servers are observable by the buyer. In reality, the buyer has to infer the service capacities from realized service times. In our study we have not considered how statistical errors may affect our results.

In Zhang's work [10], it has been shown that the multiple linear allocation can achieve the maximum feasible service capacity $\bar{\mu}$ in the Nash equilibrium, as long as server overloading and payment at customer allocation is allowed. However, as mentioned in earlier sections, server overloading and payment at allocation cause unnecessary infinite-waiting times at some out-of-equilibrium plays, and may be undesirable. It remains to be investigated whether there exists a state-independent policy without server overloading that achieves the maximum feasible service capacity. If such a policy exists, then our results would imply that such an optimal state-independent policy without server overloading (i.e. $\lambda_i \leq \mu_i$ in all allocation) also exists.

Our work has proved the existence of a n-server threshold policy that replicates any given state-independent policy that prohibits server overloading. For any fixed service capacity and given target demand allocation, it is desirable to find a mixed policy that gives the lowest expected sojourn time and randomizes between minimum number of policies. Thus, finding an efficient way to identify such a mixed policy may be a direction for future research. Since the n-server threshold policy involves a set of parameters for each service capacity vectors, another future research issue may be to investigate whether there could be simpler state-dependent policy with fewer parameters that gives the same incentives and efficiency.

Acknowledgment

Ching is supported in part by RGC Grant 7017/07P, HKU CRCG Grants, Hung Hing Ying Physical Sciences Research Fund and HKU Strategic Research Theme Fund on Computational Sciences.

References

[1] G. P. Cachon and F. Zhang, "Obtaining fast service in a queueing system via performance-based allocation of demand," Manag. Sci., vol. 53, No. 3, pp. 408–420, 2007.

[2] W. Ching, S. Choi, and M. Huang, "Optimal service capacity in a multiple-server queueing system: a game theory approach," Journal of Industrial and Management Optimization, vol. 6, pp. 73–102, 2010.

[3] W. Ching, S. Choi, and M. Huang, "Incentive effects of common and separate queues with multiple servers: the principal-agent perspective," Proceedings of the 39th International Conference on Computers and Industrial Engineering (CIE39), Troyes, France, 6-8, July, 2009.

[4] S. Choi, X. Huang, and W. Ching, Inducing optimal service capacities via performance-based allocation of demand in a queueing system with multiple servers, Working paper. 2010. http://hkumath.hku.hk/∼wkc/papers/queuepaper9.pdf

[5] S. Gilbert and Z. Weng, "Incentive effects favor nonconsolidating queues in a service system: the principal-agent perspective," Manag. Sci., vol. 44, No. 12, pp. 1662–1669, 1998.

[6] E. Kalai, M. Kamien, and M. Rubinovitch, "Optimal service speeds in a competitive environment," Manag. Sci., vol. 38, No. 8, pp. 1554 – 1163, 1992.

[7] W. P. Lin and R. Kumar, "Optimal control of a queueing system with two heterogeneous servers," IEEE Trans. Automatic Control, vol. 29, pp. 696–703, 1984.

[8] H. P. Luh and I. Viniotis, "Threshold control policies for heterogeneous server systems," Mathematical Methods of Operations Research, vol. 55, pp. 121–142, 2002.

[9] M. Rubinovitch, "The slow server problem," J. Appl. Probab., vol. 22, pp. 205–213, 1985.

[10] F. Zhang, Coordination of lead times in supply chains. Dissertation, University of Pennsylvania, Philadelphia, PA, 2004.