A Heavy Traffic Approximation for Queues with Restricted Customer-Server Matchings

(Working Paper #OM-2007-4, Stern School Business)

René A. Caldentey[†] Edward H. Kaplan[‡]

Abstract

We consider a queueing system with n customer classes and m servers. For each class i there is only a subset $S(i)$ of servers that are able to process customer' i requests and they do that using a first-come-first-serve discipline. For this system, we are primarily interested in computing P_{ij}, the steady-state fraction of class-i customers that are served by server j. We also look at stability conditions and standard performance measures like waiting times and queue lengths. Under the assumption that the system is heavy loaded, we approximate P_{ij} as well as the other performance measures. Computational experiments are used to show the quality of our approximations.

1 Introduction

There are many real-world queuing systems that pose restrictions upon which customers can be processed by which servers. Consider the following examples:

- households applying for public housing are allowed to specify those housing projects in which they are willing to live; when a public housing unit becomes newly available, of those households willing to live in the associated housing project, the one that has been waiting the longest is offered the unit (Kaplan [17])
- prospective adoptive parents specify the characteristics of infants or older children that are acceptable (e.g. age, gender, country of birth, special needs) while infants and older children with such characteristics become available for adoption at different rates; when a child becomes newly available for adoption, some mechanism for allocating this child to one set of potentially many prospective adoptive parents must be invoked (e.g. allocate to the longest waiting adoptive parent(s))

[†]Stern School of Business, New York University, New York, NY 10012, 212-998-0298, Fax: 212-995-4227, rcaldent@stern.nyu.edu

[‡]Yale School of Management and Yale School of Medicine, Yale University, New Haven, CT 06520, 203-432-6031, Fax: 203-432-9995, edward.kaplan@yale.edu

- travellers arriving at an airline counter could be processed by either a human server (further distinguished by travel class) or a computer (for customers with electronic tickets); depending upon how the queue is managed, customers with electronic tickets could be processed by either human or virtual servers, but those with paper tickets would require human assistance
- in a flexible manufacturing system, different job types require processing that can be provided by a subset of the available machines, while machines might be differentiated by the different job types they can process
- in call centers customers with different service requirements connect to the system where a set of operators with different skills and experience provide the service.

All of these examples share the basic structure that there are different types of customers and different types of servers, and that for any customer (server) type, only a subset of servers (customers) can provide (receive) service.

Of interest are basic performance characteristics for such systems. While it is relatively easy to decompose such systems if customer routing is exogenously determined (that is, if type i customers are simply assigned to type j servers with fixed probability α_{ij}), the problem becomes much more difficult when the routing is endogenously determined. In this paper, we focus on determining approximations for such systems when customer queues are processed according to FIFO within customer types, but newly available servers select among eligible customers according to FCFS across customer classes. Worded differently, whenever a server becomes free, that eligible customer who has been waiting the longest receives service. For this system, we seek to determine the fraction of type i customers who receive service from servers of type j, along with stability conditions and standard measures such as the mean queue length and waiting time.

We detail our assumptions and formally describe our model in Section 2. In Section 3 we review related literature that has addressed special cases of the system of interest. In Section 4, we consider a completely connected system where all servers can serve all customers; observations gleaned from this system motivate the heavy traffic analysis presented in Section 5, which is the heart of our paper. Section 5 includes an approximation algorithm for determining the steady-state customer flows, numerical comparisons to a simulated system, and the derivation of expected queue lengths and waiting times. We conclude in Section 6.

2 Model Description

We consider a queueing system with n customer classes and m servers. Each customer class forms a single queue, and a FIFO policy controls the output of each queue. In addition, each server uses a FCFS discipline to serve the different classes of customers. That is, if server j is free at a particular time, then from all the customers waiting for service that can be served by server j, the customer who has been waiting the longest gets served. Figure 1 shows an example with $n = 3$ and $m = 4$. In this situation, if server 1 is free, then the customer from class 1 or 3 who has been waiting the longest will start service at server 1.

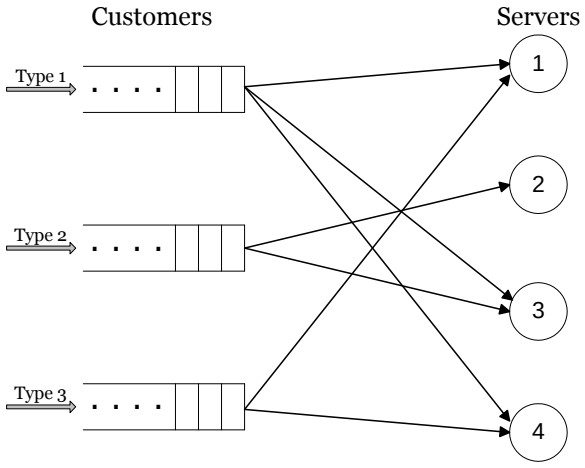


Figure 1: A simple example with 3 customer classes and 4 servers.

The data of the problem are:

1. **Dimensions**: The number of customer classes (n) and the number of servers (m). We denote by $C = \{1, 2, \ldots, n\}$ the set of all customers classes and by $S = \{1, 2, \ldots, m\}$ the set of all servers.

2. **Arrival Process**: We assume that the arrival process of class-i customers ($i \in C$) is Poisson with rate λ_i. The arrival processes for different classes are assumed to be independent.

3. **Service Process**: The service time for server j ($j \in S$) is exponentially distributed with mean μ_j^{-1}, and is independent of the customer type served.

4. **Matching Matrix**: The (0,1)-matrix $R = [R_{ij}]$ defines the "matching" of customers to servers. That is, a customer of class i can be served by server j if and only if $R_{ij} = 1$. For the example on Figure 1

$$R = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

   Based on R, we define $S(i) = \{j \in S \mid R_{ij} = 1\}$ and $C(j) = \{i \in C \mid R_{ij} = 1\}$.

5. **Customer Preferences**: If a customer can be served by more than one server at a given time, then the preference function $A(i, j, \mathcal{S})$ determines which particular server is chosen. Specifically, if a customer of class i faces the set $\mathcal{S} \subseteq S(i)$ of free servers, then with probability $A(i, j, \mathcal{S})$ the customer selects server $j \in \mathcal{S}$. A few examples of preference functions are:

   Lexicographic Preference: Within the set of available servers, always select the server with the smallest (or largest) index.
   In this case $A(i, j, \mathcal{S}) = 1$ if and only if $j \leq (\geq) k$ for all $k \in \mathcal{S}$.

RANDOM PREFERENCE: Within the set of available servers, select the server at random.

In this case $A(i, j, \mathcal{S}) = |\mathcal{S}|^{-1}$.

MINIMUM SERVICE TIME PREFERENCE: Within the set of available servers, select the server with smaller average service time (larger service rate).

In this case $A(i, j, \mathcal{S}) = 1$ if and only if $\mu_j \geq \mu_k$ for all $k \in \mathcal{S}$ (in case of a tie an additional criterion should be specified).

Without lost of generality we assume that the system is *irreducible*, that is, it cannot be decomposed into a set of completely independent queueing systems. In mathematical terms, *irreducible* means that there exists an integer $k \geq 1$ such that the $n \times n$ matrix $(R\,R')^k$ has positive elements (primes ($'$) denote vector or matrix transpose).

For this particular queueing system, we are interested in computing the fraction of class-i customers that will be served by server j. We denote this fraction by P_{ij}, and we note trivially that $P_{ij} = 0$ if $R_{ij} = 0$. Under our Markovian setting, the existence of P_{ij} is guaranteed for an ergodic system and we postpone this discussion to section §5.1. We also care about standard queueing measures such as waiting times and queue lengths.

We conclude this section with an account of some notational and terminological conventions used throughout the paper. Capital letters such as N, F, or P will be used to denote vectors or matrices; in particular, vectors are understood to be column vectors and transposes are denoted by primes. We use X_i to denote the i^{th} component of vector X and Y_{ij} for the element in row i and column j of matrix Y. Finally, if I is a set of indices and X is a vector then $X(I)$ stands for $\sum_{i \in I} X_i$, for example for $\mathcal{C} \subseteq C$, $\lambda(\mathcal{C})$ represents the cumulative arrival rate of customers in $\mathcal{C}$.

3 Related Literature

In this section we present a brief summary of the relevant. It is not our intention to review the vast literature on multi-class queueing systems. Rather, we concentrate on research that addresses the same type of specialized servers that we use in this paper.

The queueing system that we study follows in the category of queueing models with *lane selection*. It seems that lane selection was first introduced by Schwartz [20] in 1974 (see also Roque [19]), and refers to multi-server multi-class queueing systems where each customer class can be served only by a subset of the servers. Schwartz studies some examples of lane selection in a Markovian setting (*i.e.*, inter-arrival and service times are exponentially distributed) using a restrictive structure, namely, $n = m = 2$ or 3 and $R_{ij} = 1$ if and only if $j \leq i$. Expected waiting times and queue lengths are derived.

Static service (or scheduling) disciplines have been studied in a wide range of applications. Green [11]-[12] considers two customer classes (general (G) and restricted (R)), and two types of servers (general (G) and restricted (R)). There are n identical G servers and m identical R servers. R customers can only be served by R servers while G customers can be served by both types. Under a specific allocation rule ($A(i, j, \mathcal{S})$: R customers prefer R servers over G

servers) the author uses an approximation to obtain a matrix-geometric distribution for the system. Numerical results are exhibited for the expected queue lengths. The paper presents a clever way to model this particular system using only two state variables. However, the methodology does not extend easily to systems with arbitrary matching between servers and customer classes like the one we are considering in this paper. In a different setting, Kaplan [18] uses a deterministic fluid analysis to compute the waiting time when servers select at random among all eligible waiting customers. Becker *et* al. [1] study a Markovian system with C customer classes and S servers. Each arrival i is assigned to server j with fixed probability α_{ij}. The paper studies how to set the $\{\alpha_{ij}\}$ in order to minimize the total delay in the system. Glickman [10] and Filipiak [7] consider a similar static problem using deterministic fluid models.

Dynamic scheduling rules have also received some attention in the literature. One of these dynamic policies is the *generalized Shortest Queue* (e.g. Foley and Mc Donald [8], Houtum [15], Foss [9]). In this system, customers must select a server upon arrival, and they do so by joining the shortest queue. Harrison [14] and Bell and Williams [2] consider a system with two servers and two customer classes (similar to Green [11]). Under heavy traffic conditions, the authors derive a dynamic service discipline that minimizes a long-term average penalty (holding cost function) associated with the queue length.

To the best of our knowledge, there is no paper addressing the issue of lane selection under a FCFS discipline in a general multi-class multi-server setting, which is the central topic of our paper. Here we analyze the stability of the system, and present necessary and sufficient conditions for ergodicity. In addition, under heavy traffic conditions, we approximate the fraction of class-i customers served by server j (the P_{ij}'s) and standard queueing measures (average queue length and waiting time). Finally, we provide numerical examples that validate the quality of our results.

We now move to the analysis of our problem. In order to gain some intuition, we first look at the particular case of a *completely connected system*.

4 The Completely Connected Case

Suppose we have a queueing system like the one described in section §2 with the following particularity: $R_{ij} = 1$ for all $i \in C$ and $j \in S$. For obvious reasons, we call this system *completely connected*. For example, Figure 2 illustrates the queue length process for a completely connected system with two customer classes and three servers. Each node represents a state of the system. The first (and sometimes only) number inside each node is the total number (N) of customers in the system. In those cases where $N = 1$ or $N = 2$, the other number(s) represents the server(s) that is (are) working. For instance, node 2-13 describes the state where 2 customers are in the system and are being served by servers 1 and 3. In this example, class-1 customers have lexicographic preferences and class-2 customers have random preferences. We can see from the figure that for $N \geq 3$, the transitions behave according to a simple birth-death process, while for $N \leq 2$ the transitions are dictated by the preference functions and become much more irregular. We define $\pi(k, \mathcal{S})$ as the steady

state probability of being in state $N = k$ with the set $\mathcal{S}$ of servers working. Clearly, if $k \geq m$ then $\mathcal{S} = S$.

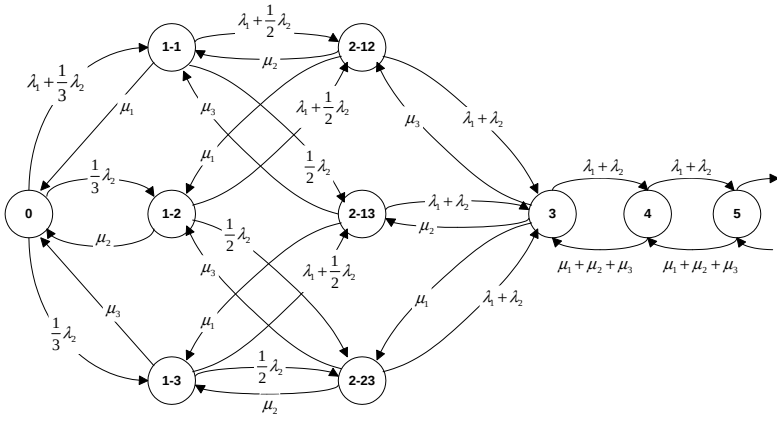


Figure 2: A completely connected system with $|C| = 2$ and $|S| = 3$. Class-1 arrivals have lexicographic preferences while class-2 arrivals have random preferences over servers.

Proposition 1 *Consider a completely connected system such that* $\sum_{k=1}^{m} \mu_k > \sum_{i=1}^{n} \lambda_i$. *Then the steady-state probabilities* $\{\pi(k, \mathcal{S})\}$ *exist and*

$$P_{ij} = \underbrace{\sum_{k=0}^{m-1} \sum_{\mathcal{S}:|\mathcal{S}|=k} \sum_{j \in \mathcal{S}^c} A(i, j, \mathcal{S}^c)\, \pi(k, \mathcal{S})}_{\textbf{(a)}} + \underbrace{\sum_{k=m}^{\infty} \left(\frac{\mu_j}{\mu_1 + \mu_2 + \cdots + \mu_m} \right) \pi(k, S)}_{\textbf{(b)}}.$$

We omit the proof of this result since it is a straightforward application of the PASTA property of the Poisson process. More interesting, however, is the distinction between the two major components (a) and (b) of the probability P_{ij}. It is not hard to see that for low-traffic systems component (a) dominates (b) and the opposite holds if the traffic intensity is high. Thus, if we set ρ^S as the traffic intensity for the system

$$\rho^S = \frac{\lambda_1 + \lambda_2 + \cdots + \lambda_n}{\mu_1 + \mu_2 + \cdots + \mu_m}$$

then for $\rho^S \to 1^-$ we get

$$P_{ij} \approx \frac{\mu_j}{\mu_1 + \mu_2 + \cdots + \mu_m}. \tag{1}$$

The intuition behind this result is simple. If all servers are working continuously, then during a time period of length T the total output of the system is approximately $(\mu_1 + \mu_2 + \cdots + \mu_m)T$ and the total output of server j is approximately $\mu_j T$. Thus, the fraction of the total output produced by server j is equal to the right hand side in (1). We note that the solution in (1) is independent of the allocation policy $\{A(i, j, \mathcal{S})\}$. The reason is that for a system with high utilization, it is very unlikely that an arriving customer will find more than one server idle.

On the other hand, if the traffic intensity is very low, $\rho^S \to 0^+$, we get $\pi(0, \emptyset) \approx 1$ and thus

$$P_{ij} \approx A(i, j, S). \tag{2}$$

In this case, the allocation rule $\{A(i, j, S)\}$ plays a major role in the representation of P_{ij}.

While the case of moderate traffic intensity ($0 < \rho^S < 1$) is the most general, closed-form analysis of this situation seems prohibitive even for the very simple case of a completely connected system. In addition, we believe that systems with high utilization require much attention and control. For these reasons, and also to maintain tractability, we restrict our study of the P_{ij}'s to the extreme scenario $\rho^S \to 1^-$.

We note that for the other extreme case $\rho^S \to 0^+$ condition (2) will remain true for general systems.

5 Heavy Traffic Analysis

In this section, we consider a general Markovian system with n customer classes, m servers, and an arbitrary irreducible matching matrix R. We assume that the system operates under heavy traffic conditions, that is, given $\epsilon \geq 0$ small the system utilization ρ^S satisfies

$$1 > \rho^S = \frac{\lambda_1 + \lambda_2 + \cdots + \lambda_n}{\mu_1 + \mu_2 + \cdots + \mu_m} \geq 1 - \epsilon.$$

In this situation, we expect queues to be large and servers to be working continuously. For this system, we will study (i) the "assignment" of customers to servers (the P_{ij} probabilities) as well as (ii) waiting time and queue lengths for the different classes.

In the heavy-traffic regime, the fact that queues are (almost) never empty implies that the values of the P_{ij}'s remain somehow invariant to the values of the traffic intensity as long as the servers are never (or almost never) idle; see §5.22 below. So stability considerations, such as $1 > \rho^S$, are not critical in our analysis of the P_{ij}'s.

On the other hand, performance measures of the system like queue lengths and waiting time are affected by stability considerations. Certainly, $\rho^S < 1$ is necessary to ensure ergodicity, however, it is far from being sufficient for the general case. As long as we have a class i of customers such that $S(i) \neq S$ the study of stability requires more than just the analysis of the aggregate traffic intensity (ρ^S). We deal with this issue of stability in the following section.

5.1 Stability Considerations

For expository purposes, we present a necessary and sufficient condition for stability (in the sense of *positive Harris recurrence*, Dai [5]) using a modified system. Specifically, we consider a system where customers must select a server upon arrival, and they cannot change this decision later as the service process evolves. We assume that customers select servers in a Markovian fashion, *i.e.*, there are fixed nonnegative quantities α_{ij} representing the

probability that a class-i customer selects server $j \in S(i)$. We will refer to this modified system as RQS (*Random Queue Selection*) and to our original system as SQT (*Shortest Queueing Time*).

By definition, the SQT model is work conserving since no server j is idle if there is a customer $i \in C(j)$ waiting for service. On the other hand, the RQS model does not satisfy this property because of the probabilistic assignment. For example, a customer can select a nonempty queue even if there is an empty server that can serve him/her. In this respect, the RQS model is inefficient on the usage of service capacity. This observation leads us to argue that stability of the RQS model implies stability of the SQT model. Let us then look at the stability of the RQS system.

By construction, the RQS model decouples the system into m independent M/M/1 queues. The arrival rate for system j ($j = 1, \ldots, m$) is equal to $\sum_{i \in C(j)} \alpha_{ij} \lambda_i$ and the service rate is μ_j. The RQS model is stable if each of the m M/M/1 systems is stable which occurs if and only if the arrival rate is strictly less than the service rate, *i.e.*,

$$\sum_{i \in C(j)} \lambda_i \alpha_{ij} < \mu_j \qquad \text{for all } j = 1, \ldots, m. \tag{3}$$

Given our previous discussion, we can argue that a sufficient condition for the stability of the original system SQT is the existence of probabilities $\{\alpha_{ij}\}$ that satisfy (3). Using an LP argument (specifically a Max-Flow formulation) as in Foley and McDonald [8] , it can be shown that (3) is equivalent to the following condition

$$\lambda(B) = \sum_{i \in B} \lambda_i < \sum_{j \in S(B)} \mu_j = \mu(S(B)) \qquad \text{for all } B \subseteq S. \tag{4}$$

Intuitively, this condition requires that for any subset B of customer classes the total arrival rate $\lambda(B)$ of this set has to be smaller than the total service capacity $\mu(S(B))$ available to serve the customers in B. Certainly, this condition (4) is necessary for stability, however, given its equivalence to (3) it turns out to be also sufficient. In summary, we have the following result.

Proposition 2 *The queueing system (SQT) under consideration is stable if and only if condition (4) is satisfied.*

Proof: see the appendix at the end.

5.2 Computing the Matching Distribution P_{ij}

We now turn to the analysis of the P_{ij}'s. As we will see shortly, under heavy traffic conditions, the problem of computing the P_{ij}'s can be formulated as a matching problem in an infinite dimensional bipartite random graph. This formulation is useful to understand the inherent complexity of the problem which unfortunately we have not been able to solve in closed form. Instead, we propose an approximation based on a simple perturbation idea.

5.2.1 Acyclic Systems and Bounds

Before looking at the general case, let us discuss briefly those cases where closed-form solutions for the P_{ij}'s are available and how we can in general get bounds for these quantities.

Suppose at time $t = 0$ we start with an empty system and let $\tau > 0$ be the busy period. That is, τ is the time it takes for the system to empty after serving at least one customer. Let $N_i^a(\tau)$ be the total number of class i customers arriving during the busy period and $N_j^s(\tau)$ be the total number of service completions by server j during the same period. We also set $F_{ij}(\tau)$ to be the total number of customers i that end up being served by sever j. Since at τ the system is empty we must have that

$$\begin{aligned} \sum_{j \in S(i)} F_{ij}(\tau) &= N_i^a(\tau) \qquad \text{for all } i = 1, \ldots, n \\ \sum_{i \in C(j)} F_{ij}(\tau) &= N_j^s(\tau) \qquad \text{for all } j = 1, \ldots, m. \end{aligned}$$

As the traffic intensity of the system ρ^S goes to one the busy period τ goes to infinity w.p.1. In this regime, if the stability condition (4) is asymptotically satisfied then all servers are working continuously and by the law of large numbers we have that $N_i^a(\tau) \sim \lambda_i \tau$, $N_j^s(\tau) \sim \mu_j \tau$, and $F_{ij}(\tau)/N_i^a(\tau) \sim P_{ij}$. Thus, in the limit as $\rho^S \to 1^-$ the previous system asymptotically implies that

$$\sum_{j \in S(i)} P_{ij} = 1 \qquad \text{for all } i = 1, \ldots, n \tag{5}$$

$$\sum_{i \in C(j)}^{n} \lambda_i P_{ij} = \mu_j \qquad \text{for all } j = 1, \ldots, m. \tag{6}$$

In addition, we want the P_{ij} to be non-negative, that is,

$$P_{ij} \geq 0 \qquad \text{for all } i = 1, \ldots, n \; j = 1, \ldots, m. \tag{7}$$

We refer to (5),(6), and (7) as the heavy traffic balance equations. We note that (5) and (6) imply $\rho^S = 1$ which is the heavy traffic condition that we will use for computing the P_{ij}'s. At this point the reader might be concerned about the stability of our queueing system under this heavy traffic condition. However, this concern is irrelevant, as we will see on §5.2.2, since our computation of the P_{ij}'s is based on the requirement that servers are never idle w.p.1. which clearly holds if the system (5)-(7) is feasible.

The feasibility of this system (5)-(7) is guaranteed if (4) is asymptotically satisfied as $\rho^S \to 1^-$. However, the solution is not necessarily unique. Uniqueness is achieved under some restrictive condition on the matching matrix R. For example, Figure 3a shows a case where the solution of (5)-(6) is unique as $\rho^S \to 1^-$ ($\epsilon \to 0^+$). For this example the P_{ij}'s tend to $P_{11} = 1$, $P_{21} = P_{22} = 0.5$, and $P_{33} = 2\,P_{32} = 0.66\overline{6}$. Therefore, when there is limited flexibility to assign customers to servers, the computation of the P_{ij}'s becomes trivial in the heavy traffic regime. We refer to systems that fall into this category as *acyclic*[†]. In order

[†]In a dynamic scheduling setting, Harrison [14] and Bell and Williams [2] studied a two-server acyclic system.

to give a formal characterization of these acyclic systems, we view the sets of customers and servers as nodes in an undirected bipartite graph where the incidence matrix is defined by the matrix R (see Figure 3b). We say that the queueing system is acyclic if the corresponding graph is acyclic. The definition of an acyclic system leads to the following result:

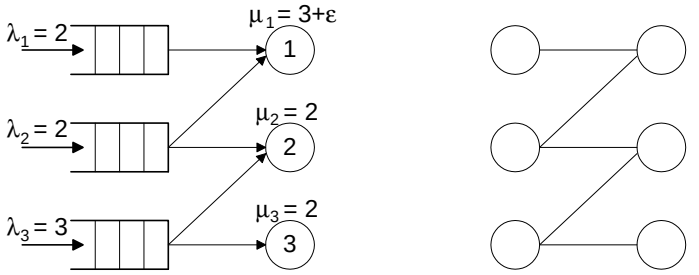
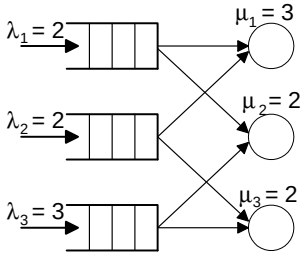


Figure 3: (a): A simple example where (5)-(6) has a unique solution as $\epsilon \to 0^+$. (b): Undirected graph associated to the queueing system in (a).

Proposition 3 *If the queueing system is acyclic and stable then P_{ij} is the unique nonnegative solution of the system (5)-(6) in the heavy traffic regime.*

The proof follows directly from our previous discussion and the fact that the graph associated to an acyclic system is a simple spanning tree.

In the cases that (5)-(6) does not possess a unique solution, we can still use this linear system to obtain upper and lower bounds. Let $\mathcal{P} = \{P \geq 0 : P \text{ satisfies } (5) - (6)\}$ be the polyhedron of feasible solutions. Then, the linear programs $\min_{P\in\mathcal{P}}\{P_{ij}\}$ and $\max_{P\in\mathcal{P}}\{P_{ij}\}$ produce lower and upper bounds for the value of P_{ij}, respectively. For example, consider the queueing system in Figure 4. In this case (5)-(6) does not have a closed form solution. The table on the right of the figure compares the values of the upper and lower bounds with estimates obtained using simulations. In general, the quality of these bounds depends heavily



P_{ij}	Bounds		Simulation
	Upper	Lower	
P_{11}	1	0.5	0.75
P_{12}	0.5	0	0.25
P_{21}	1	0.5	0.75
P_{23}	0.5	0	0.25
P_{32}	0.667	0.333	0.5
P_{33}	0.667	0.333	0.5

Figure 4: For the queueing system on the right (5)-(6) does not have unique solution. The table on the left shows the upper and lower bounds for P_{ij} as well as the values obtained using simulation.

on the topology of the system (matrix R). In particular, it is not hard to find examples where the bounds do not provide any information, *i.e.*, the upper bound is 1 and the lower bound is 0 for all P_{ij}. However, in some cases they can provide some guidance about the behavior of the P_{ij} like in the example of Figure 4.

5.2.2 Matching Formulation

We now formulate the problem of computing the P_{ij} as a matching problem in an infinite dimension bipartite random graph. This formulation is not particularly useful for solving the problem. However, it provides an intuitive mathematical representation of the problem and reveals the symmetry between the arrival and service processes (under heavy traffic conditions) that we will exploit later. The only assumption that we use in this exposition is that servers are almost never idle w.p.1 which is essentially our heavy traffic condition.

Suppose that we start with a system in the following condition: (i) all servers are working and (ii) if a particular customer c is in the queue then all the customers that have arrived after c are also waiting on queue. Condition (ii) is not really needed but makes the exposition easier, we only require condition (i) to hold. Take the customers on the queue plus all future arrivals and order them according to their arrival time starting from the earliest arrival. We denote this sequence by $a = (a_1, a_2, \ldots)$, where $a_k \in C$ is the type of the k^{th} member in the sequence. Under condition (ii) the $\{a_k\}$ are i.i.d. multinomial random variables such that $\Pr(a_k = i) = \lambda_i/\lambda(C)$. Similarly, let consider the sequence $s = (s_1, s_2, \ldots)$ of service completions, where $s_k \in S$ is the server that will complete the k^{th} service. The $\{s_k\}$ are also i.i.d. multinomial random variable with p.m.f. $\Pr(s_k = j) = \mu_j/\mu(S)$.

Under condition (i) every time a server gets free there is at least one member of the sequence a that is waiting in line for service. If this is the case, the only information required to study the P_{ij}'s is the sequence of arrival (a), the sequence of service completions (s), and the matrix R. In this setting, the assignment of customers to servers is equivalent to matching the elements in the sequence of arrivals to the elements in the sequence of services. Let us illustrate this matching with the following example. Figure 5b shows how the matching

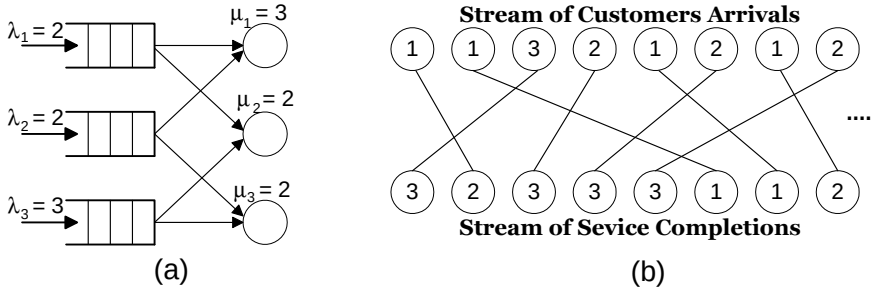


Figure 5: An example of the customer-server matching. Figure (a) shows the queueing system. Figure (b) illustrates how the matching is done for an specific stream of arrivals and service completions.

between customers and servers is done (for the system in Figure 5a) once the streams a and s are defined. We notice that the roles played by customers and servers are perfectly symmetric, that is, we can interchange the streams of customers and service completions and the resulting matching remains the same. In addition, in this heavy traffic regime the absolute value of the service and arrival rates does not affect the matching what really matters is the probability distributions of the sequences a and s which depend on their relative value. We can formally define the matching as a mapping $\mathcal{M} : a \to s$ such that $\mathcal{M}(a_k) = s_t$ if and only if the following two conditions are satisfy.

1. $s_t \in S(a_k)$.

2. For all $\tilde{k} < k$ such that $s_t \in S(a_{\tilde{k}})$ it must be that $\mathcal{M}(a_{\tilde{k}}) = s_{\tilde{t}}$ with $\tilde{t} < t$.

Given this matching $\mathcal{M}$, the value of P_{ij} is given by

$$P_{ij} = \lim_{K\to\infty} \frac{\sum_{k=1}^{K} \mathbb{1}(a_k = i)\ \mathbb{1}(\mathcal{M}(a_k) = j)}{\sum_{k=1}^{K} \mathbb{1}(a_k = i)},$$

where $\mathbb{1}(X)$ is the indicator function of event X. Computing this limit above is a task that we have not been able to do. The major difficulty is the path-dependent nature of the mapping $\mathcal{M}$. That is, to verify the value of $\mathbb{1}(\mathcal{M}(a_k) = j)$ the whole history $(a_1, \ldots, a_k)$ and the full sequence s are essentially needed and we have not been able to summarize in a tractable way this information. For this reason we approach the problem using an approximation that ends up producing good results.

5.2.3 Approximations

The approximation that we propose is based on a property of the *completely connected* systems as $\rho^S \to 1^-$. In this simplify scenario, we have that the fraction of customers i that select server j is simply

$$P_{ij} = \frac{\mu_j}{\mu_1 + \mu_2 + \cdots + \mu_m},$$

as $\rho^S \to 1^-$. That is, P_{ij} is the proportion of the total service capacity that is offered by server j. Since this result holds under heavy traffic condition, from the rest of this section §5.2.3 we will consider only the asymptotic regime of a stable system, that is, the system (5)-(7) admits a feasible solution.

The behavior of P_{ij} for a completely connected system leads us to the following observation:

Observation: *In the absence of any type of interference from the other classes of customers, the fraction of customers i assigned to server $j \in S(i)$ would be given by*

$$\frac{\mu_j\, R_{ij}}{\sum_{k\in S(i)} \mu_k} \tag{8}$$

under heavy traffic conditions.

Of courses, servers in $S(i)$ are not exclusively assigned to class-i customers and (8) is only a crude approximation for the true values of the P_{ij}'s. Moreover, there is not guarantee that the they satisfy at least the balance conditions (5)-(6). What is important about (8) is that it reflects a natural tendency to allocate customers to servers. The key idea of our approximation is to construct an assignment that is feasible (satisfies (5)-(6)) and at the same time uses as much as possible the allocation proposed by (8). The discussion of this approximation is made in terms of the flows $F_{ij} = \lambda_i\, P_{ij}$ rather than the probabilities P_{ij}.

Consider a *completely connected* system and let F_{ij}^C be the average flow of customers i assigned to server j, that is,

$$F_{ij}^C = \frac{\lambda_i\, \mu_j}{\mu_1 + \ldots + \mu_m}.$$

Note that by construction, the F^C_{ij} flows balance the arrival rate (input) and the service rate (output) meaning that $\sum_i F^C_{ij} = \mu_j$. However, the F^C flows are only an "ideal" assignment since some of the arcs (i, j) do not exist in our original system. Associated to this "ideal flow" F^C we define three quantities: (i) F^0 the feasible part of F^C, (ii) E^0_i the residual flow of customers i, and (iii) D^0_j the residual flow at server j by:

$$F^0_{ij} = F^C_{ij}\, R_{ij}, \quad E^0_i = \lambda_i - \sum_{j=1}^{m} F^0_{ij}, \text{ and } \quad D^0_j = -(\mu_j - \sum_{i=1}^{n} F^0_{ij}). \tag{9}$$

We interpret this quantities as follows. Suppose that we start with a perfectly connected system. In this case, we have that $F_{ij} = F^C_{ij}$. Suppose then that suddenly we block all the flows (i, j) for which $R_{ij} = 0$ (i.e., $F_{ij} = 0$). In this case, we have that instantaneously the rate of customers i that get blocked is E^0_i and the amount of idle capacity at server j is equal to $-D^0_j$. From this observation and our assumption $\rho^S = 1$, it is clear that $\sum_{i=1}^n E^0_i + \sum_{j=1}^m D^0_j = 0$. We introduce two auxiliary matrices L_{ij} and M_{ij} given by

$$L_{ij} = \frac{\lambda_i\, R_{ij}}{\sum_{k\in C(j)} \lambda_k} \qquad\qquad M_{ij} = \frac{\mu_j\, R_{ij}}{\sum_{k\in S(i)} \mu_k}.$$

L_{ij} represents the fraction of load imposed by class-i customers to server j while M_{ij} represents the fraction of capacity offered by server j to class-i customers. Notice that M_{ij} is equal to the allocation described in (8). According to our observation matrix M define a natural way to allocate customers to servers. Similarly, matrix L defines a natural way to allocate service capacity to customers. This observation follows from the symmetry between service and arrival processes.

We are now ready to explain our proposed solution. The approach is to construct a sequence of systems such that system k is characterized by three quantities (i) an $(n \times m)$ matrix of flow rates $F^k = [F^k_{ij}]$, (ii) an n-vector of residual input rates $E^k = (E^k_i)$, and (iii) a m-vector of residual output rates $D^k = (D^k_j)$. In the limit as $k \to \infty$, F^k_{ij} will converge to our proposed solution. For the sake of clarity we will complement the analytical derivation with the following simple 4×3 example in figure 6. Given this system, we first assume that we have a *perfectly connected* system and compute F^C_{ij}. Then, since not all the arcs are present we compute the residual vectors E^0_i and D^0_j. We set our initial condition to be (F^0_{ij}, E^0_i, D^0_j). Figure 7 shows the resulting flows.

Given this initial situation, our proposed solution is constructed as follows:

$$E^{k+1} = L\, D^k \tag{10}$$

$$D^{k+1} = M'\, E^k \tag{11}$$

$$F^{k+1}_{ij} = F^k_{ij} + M_{ij}\, E^{k-1}_i - L_{ij}\, D^k_j \quad k \geq 1. \tag{12}$$

Intuitively, what this solution does at every stage is to allocate the "unassigned customers" (E^k) to the servers using the allocation rule induced by matrix L. Similarly, the idle "capacity" (D^k) is distributed among the customers using matrix M'. This successive allocation of customers to servers and servers to customers generates a sequence of flows resulting in

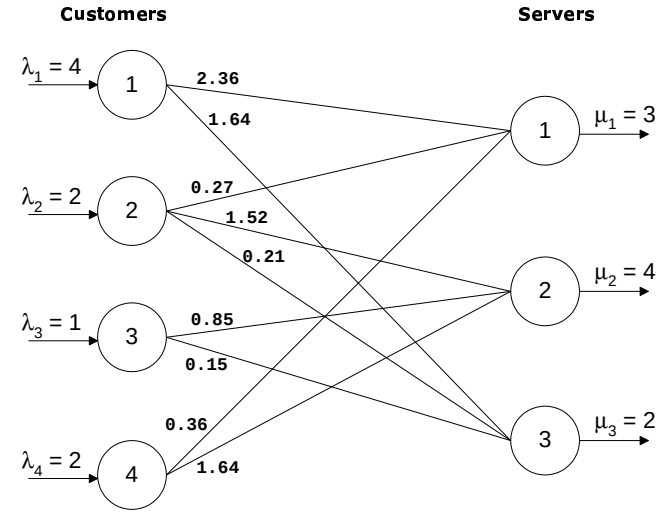


Figure 6: A 4×3 example. The number on the arcs are the values of the flows F_{ij} obtained using simulation.

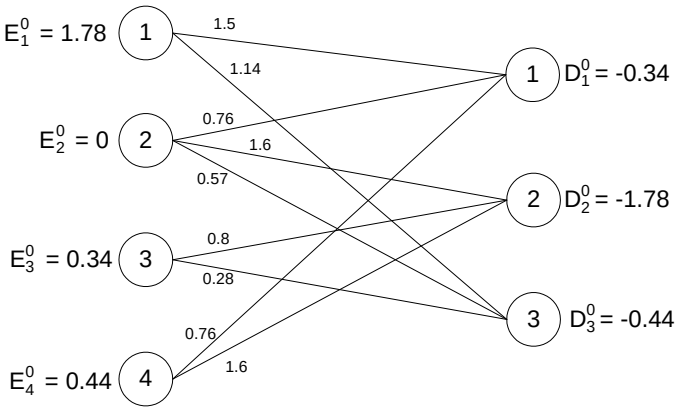


Figure 7: The number on the arcs are the values of the flows F^0_{ij}.

condition (12). The other two conditions (10) and (11) characterize the way the successive residual input and residual output vectors are generated. At least two things need to be proven to ensure that the sequence of flows generated by (12) is a good candidate for an approximations. One is the convergence of the sequence and the other is its consistency with (5)-(6).

We start proving convergence. For this purpose, we redefine (10)-(12) using an equivalent representation which is easy to work with. Let $N^0 = E^0 + L\,D^0$, $\widetilde{F}^0_{ij} = F^0_{ij} - L_{ij}\,D^0_j$, and $A = L\,M'$ then the following conditions are equivalent to the system (10)-(12).

$$N^{k+1} = A\,N^k \tag{13}$$

$$F^{k+1}_{ij} = \widetilde{F}^0_{ij} + M_{ij}\sum_{n=0}^{k} N^n_i - L_{ij}\sum_{n=0}^{k}(M'\,N^n)_j. \tag{14}$$

Two properties of A are useful. From the definition of L and M it easy to see that $A = L\,M'$

is a $(n \times n)$ stochastic matrix such that $\sum_{i=1}^{n} A_{ij} = 1$ for all $j = 1, \ldots, n$. Then, from the Perron-Frobenius theory we have that A^k converges to a stochastic matrix A^∞ such that the elements in a given row are all equal, *i.e.*, $A^\infty_{ij} = A^\infty_{ik}$ for all $i, j, k = 1, \ldots, n$. In addition, the following preliminary result is needed.

Lemma 1 *For an irreducible system, there is a n-dimensional vector W such that*

$$W = N^0 + A\,W. \tag{15}$$

Proof: We need to prove that $N^0 \in \prec I - A \succ$, where $\prec I - A \succ$ is the linear space generated by the matrix $I - A$. Since the system is irreducible matrix A can be view as the transition matrix for a single recurrent Markov chain then $\dim(\prec I - A \succ) = n - 1$, *i.e.*, the linear space generated by $I - A$ has dimension $n - 1$. We define the linear space $\mathcal{Z} = \{X \in \Re^n : \sum_{i=1}^{n} X_i = 0\}$. We notice that $\dim(\mathcal{Z}) = n - 1$ and $N^0 \in \mathcal{Z}$ by construction. Finally, since A is a transition matrix $\prec I - A \succ \subseteq \mathcal{Z}$. But both $\prec I - A \succ$ and $\mathcal{Z}$ have the same dimension, we conclude that $\prec I - A \succ = \mathcal{Z}$ and $N^0 \in \prec I - A \succ$. ∎

We are now ready to prove the convergence of F^k_{ij}.

Proposition 4 *The sequence $\{F^k_{ij}\}_{k\geq 0}$, defined in (14), converges for every pair (i, j).*

Proof: By lemma (1) above we know that there exists a vector W such that $W = N^0 + A\,W$. If we iterate this identity k times we get

$$W = N^0 + A\,N^0 + A^2\,N^0 + \cdots + A^k\,N^0 + A^{k+1}\,W \Longrightarrow \sum_{n=0}^{k} A^n\,N^0 = W - A^{k+1}\,W.$$

Therefore, we can rewrite (14) as follows.

$$F^{k+1}_{ij} = \widetilde{F}^0_{ij} + M_{ij}\left(W - A^{k+1}\,W\right)_i - L_{ij}\left(M'(W - A^{k+1}\,W)\right)_j.$$

We have already argued that $\lim_{k\to\infty} A^k$ exits and we have called this limit A^∞. Thus, letting $V = W - A^\infty W$ we get

$$F_{ij} \equiv F^\infty_{ij} = \widetilde{F}^0_{ij} + M_{ij}\,V_i - L_{ij}\,(M'\,V)_j \tag{16}$$

which completes the proof. ∎

The second condition that we need to check is the consistency of F_{ij} with (5)-(6).

Proposition 5 *In the heavy traffic regime $\rho^S = 1$ the limit flows F_{ij} in (16) satisfy the following conditions:*

$$\sum_{i=1}^{n} F_{ij} = \mu_j \qquad \sum_{j=1}^{m} F_{ij} = \lambda_i.$$

Proof: See the appendix at the end.

5.2.4 An Alternative Characterization of the Equilibrium Flows

Another approach to determining the flows F_{ij} proceeds by noting the form of the solution in equation (16). Let

$$X_j \equiv (M' V)_j. \tag{17}$$

Then the flows may be written as

$$F_{ij} = \widetilde{F}^0_{ij} + M_{ij}V_i - L_{ij}X_j \tag{18}$$

where $\widetilde{F}^0_{ij}$, M_{ij} and L_{ij} are known, but V_i and X_j are treated as unknown quantities. In other words, we proceed by representing the equilibrium flows in the *form* given by equation (18). However, the flows must satisfy the balance requirements of Proposition 5, thus one can *select* V_i and X_j to solve the equations

$$\sum_{i=1}^{n}(\widetilde{F}^0_{ij} + M_{ij}V_i - L_{ij}X_j) = \mu_j \text{ and } \sum_{j=1}^{m}(\widetilde{F}^0_{ij} + M_{ij}V_i - L_{ij}X_j) = \lambda_i. \tag{19}$$

There is one extra degree of freedom since in the heavy traffic regime $\sum_j \mu_j = \sum_i \lambda_i$, so we need an additional equation to determine $m + n$ parameters (the V_i's and X_j's) from the flow balance equations. For example, adding

$$\sum_{i=1}^{n} V_i = 0 \tag{20}$$

to (19) is a convenient choice. It is easy to establish that once the parameters V_i and X_j have been determined as suggested, the equilibrium flows that result from equation (18) are identical to those that result from (16).

While we have shown that our limit flows satisfy the heavy traffic balance equations (5) and (6), we have not been able to prove that the resulting flows are non-negative (as they must be for a feasible system). However, in the next section we present numerical evidence indicating that our limit flows are not only feasible; they provide very close matches to simulated flows in heavy traffic.

5.2.5 Computational Experiments

This section reports computational experiments that show the quality of our proposed solution. Three major elements are important when designing the experiments:

1. The dimension of the problem in terms of n and m.
2. Traffic intensity ρ^S. Given that our methodology is appropriate only for heavy loaded system, we will consider instances satisfying $\rho^S \geq 0.9$.
3. Density of the matching matrix R. We define the density

$$\nu(R) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} R_{ij}}{n\,m},$$

i.e., the fraction of nonzero entries. Notice that by changing the density of a system we are implicitly changing the routine structure.

Table 1: Values of ϕ for a system with dimensions $n = 3$ and $m = 5$.

ρ^S	$\nu(R) = 0.6$	$\nu(R) = 0.73$	$\nu(R) = 0.87$	$\nu(R) = 0.93$
1.00	0.0006	0.0097	0.0016	0.0011
0.99	0.0046	0.0105	0.0016	0.0011
0.96	0.0152	0.0162	0.0022	0.0022
0.93	0.0267	0.0159	0.0027	0.0029
0.90	0.0404	0.0222	0.0033	0.0045

We also require a measure that defines the quality of the approximation. Suppose that P_{ij}^S is the solution of the simulation and P_{ij}^* is the solution obtained using (16), *i.e.*, $P_{ij}^* = F_{ij}/\lambda_i$. Then, we define the measure ϕ as follows

$$\phi = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} |P_{ij}^S - P_{ij}^*|}{n\, m\, \nu(R)}.$$

That is, ϕ computes the average absolute difference between P_{ij}^S and P_{ij}^*. The following tables present the results for three different set of dimensions.

Table 2: Values of ϕ for a system with dimensions $n = 7$ and $m = 5$.

ρ^S	$\nu(R) = 0.54$	$\nu(R) = 0.66$	$\nu(R) = 0.86$	$\nu(R) = 0.94$
1.00	0.0024	0.0076	0.0031	0.0023
0.99	0.0037	0.0071	0.0034	0.0023
0.96	0.0079	0.0083	0.0056	0.0034
0.93	0.0087	0.0135	0.0079	0.0043
0.90	0.0106	0.0172	0.0093	0.0055

Table 3: Values of ϕ for a system with dimensions $n = 10$ and $m = 7$.

ρ^S	$\nu(R) = 0.47$	$\nu(R) = 0.63$	$\nu(R) = 0.8$	$\nu(R) = 0.96$
1.06	0.0075	0.0035	0.0018	0.0016
1.00	0.0063	0.0041	0.0021	0.0018
0.99	0.0067	0.0042	0.0019	0.0018
0.96	0.0086	0.0047	0.0028	0.0026
0.93	0.0112	0.0063	0.0039	0.0040
0.90	0.0157	0.0086	0.0049	0.0054

As a general comment, we can see that the solution proposed by (16) is quite accurate. In general the average error is below 0.01, that is, the proposed solution is able to predict the values of P_{ij} with an absolute error less than 1%. Moreover, we notice that we are comparing the quality of (16) using the solution of the simulation which is not certain to be 100% exact. Another observation is related to the performance of the proposed solution with the traffic intensity ρ^S. As we might expect the performance deteriorates as ρ^S decreases, although slightly in the range $\rho^S \geq 0.9$. In conclusion, we have provided a closed-form representation of the P_{ij} that seems to match almost identically the results of the simulation. This solution performs very well in the range $\rho^S \geq 0.9$ improving as ρ^S increases.

5.3 Performance Measures

We compute average queue length and waiting time under the assumption that the system is heavy loaded $\rho^S \to 1^-$ and stable in the limit, *i.e.*, conditions (5)-(6) admit a nonnegative feasible solution for $\rho^S \to 1^-$. Following closely the work by Borokov [4] and Iglehart and Whitt [16], we heuristically derive the steady-state distribution of the total queue length in the system (the total number of customers of all classes in the system) and an estimate of the average waiting for each class.

The setting in Iglehart and Whitt [16] is very similar to ours. There are n customers classes and m different servers. Every customer joins the same queue and every server can serve any customer. A FCFS service discipline is used. The main difference in our case is that a particular server can serve only a subset of the customers' classes. It should be clear that in our systems queue lengths and waiting time are larger than in the Iglehart and Whitt model. However, these differences become negligible in the heavy traffic regime. In order to see this, let us first introduce some notation.

- Let $\{A_k^i,\ k \geq 1\}$ be the sequence of inter-arrival time for class-i customers ($i = 1, \ldots, n$).
- Let $\{S_k^j,\ k \geq 1\}$ be the sequence of service time for server j ($j = 1, \ldots, m$).
- For simplicity we assume that the system starts empty.

Given this primitive data the queue length and waiting processes are fully determined.

In order to prove our result we will slightly modify the service process in order to transform our system into one that looks like the Iglehart and Whitt system. For this purpose consider the sequence of departure epochs of server j described in Figure 8. The diagram on top describes the sequence of services and idle period for server j in our system. The idle period of server j can be divided in two categories:

Type 1 idle periods are those that start when server j finishes a service and (i) there are no more customers on queue that can be served by server j but (ii) there are customers of other classes on queue. We denote this sequence of idle periods by $\{\hat{I}_k^j,\ k \geq 1\}$.

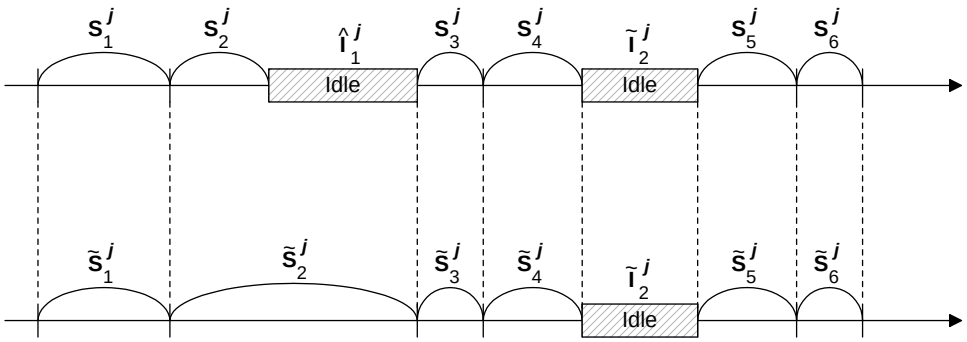


Figure 8: Sequence of busy and idle period for server j.

Type 2 idle periods are those that start when server j finishes a service and there are no more customers on queue of any type. We denote this sequence of idle periods by $\{\tilde{I}_k^j,\ k \geq 1\}$.

In the Iglehart and Whitt model Type 1 idle periods do not happen since all servers are able to serve all classes. Type 2 idle periods are obviously common to both systems. Let us define

$$\mathcal{K}^j \stackrel{\text{def}}{=} \{k \geq 1 \text{ s.t. } S_k^j \text{ immediately precedes a type 1 idle period for server } j\}.$$

In the example of Figure 8 we have that $2 \in \mathcal{K}^j$ but $4 \notin \mathcal{K}^j$. In order to avoid type 1 idle periods for server j we modify the service time S_k^j for all $k \in \mathcal{K}^j$ and define a new sequence of service time $\{\tilde{S}_k^j,\ k \geq 1\}$ as follows.

$$\tilde{S}_k^j = \begin{cases} S_k^j & \text{if } k \notin \mathcal{K}^j \\ S_k^j + \hat{I}_{\hat{k}}^j & \text{if } k \in \mathcal{K}^j, \end{cases}$$

where $\hat{I}_{\hat{k}}^j$ is the type 1 idle period that immediately follows after S_k^j, $k \in \mathcal{K}^j$. Figure 8 shows how to construct the modified sequence of service time $\{\tilde{S}_k^j,\ k \geq 1\}$. The sequences $\{A_k^i\}$ and $\{\tilde{S}_k^j\}$ define a new queueing system that satisfied the condition that no server is idle as long as there are customers in queue. Hence with this artificial construction, we are able to apply the results of Iglehart and Whitt. The only obstacle that we have is the fact that the sequences $\{\tilde{S}^j\}$ are not i.i.d. This problem is, however, irrelevant in the heavy traffic regime under consideration since the steady-state probability of Type 1 idle periods goes to zero as $\rho^S \to 1^-$. Thus, let $\tilde{Q}(t)$ be the aggregate queueing process (total number of customers in queue plus on service for all classes) for the modified system. For a large number $h > 0$ (the scaling factor), we define the scaled drift and queueing process as

$$\theta_h = \sqrt{h}(\rho^S - 1)\mu(S) \quad \text{and} \quad \tilde{Q}_h(t) = \frac{\tilde{Q}(ht)}{\sqrt{h}}, \text{ respectively.}$$

Suppose that the traffic intensity $\rho^S \to 1^-$ in such a way that $\lim_{h\to\infty} \theta_h = \theta < 0$. Then, it follows from Iglehart and Whitt [16] and our previous discussion that $\tilde{Q}_h(t) \Rightarrow |\xi(t)|$ if

$\rho^S \to 1^-$, where ξ is a Brownian motion with drift θ and variance $\sigma^2 = \lambda(C) + \mu(S)$[†] (where $\Rightarrow$ stands for weak convergence on the space of *cádlág* functions, see Billingslley [3]).

Let $Q(t)$ be the aggregate queueing processes for the original system. Then, given our construction of the modified queue process $\tilde{Q}(t)$ it follows that

$$\tilde{Q}(t) - m \leq Q(t) \leq \tilde{Q}(t).$$

Therefore, under a heavy traffic scaling we have that

$$\tilde{Q}_h(t) - \frac{m}{\sqrt{h}} \leq Q_h(t) \leq \tilde{Q}_h(t). \tag{21}$$

This condition together with the convergence of $\tilde{Q}_h(t)$ imply that $Q_h(t) \Rightarrow |\xi(t)|$, a (θ, σ)-Brownian motion. Moreover, it is well-known that such one-sided RBM have a negative exponential distribution with mean $\sigma^2/2|\theta|$ (*e.g.*, Harrison [13], section §5.6). From this result we can approximate the expected total number of customers in the system by

$$Q^\infty = \lim_{t\to\infty} E[Q(t)] \approx \frac{\lambda(C) + \mu(S)}{2(\mu(S) - \lambda(C))} = \frac{1 + \rho^S}{2(1 - \rho^S)}.$$

In addition, in the heavy traffic regime we expect the steady-state aggregate queue length to have an homogeneous composition in terms of customers classes. That is, if $Q_i(t)$ is the queue length process of class i (so that $Q(t) = \sum_{i=1}^n Q_i(t)$) then we expect that

$$Q_i^\infty = \left(\frac{\lambda_i}{\lambda(C)}\right) E[Q(\infty)] = \frac{\lambda_i(1 + \rho^S)}{2\lambda(C)(1 - \rho^S)} \quad \text{for all } i = 1, \ldots, n. \tag{22}$$

This equality, however, is not guaranteed because in our system there is over-taking. However, in the heavy traffic regime queues are large and over-taking is only affecting those customers in queue that are closed to the server. Hence, the big majority of the customers in the queue at any moment of time have not experienced any over-taking service and we expect (22) to hold for $\rho^S \to 1^-$. Finally, from (22) and Little's law we can estimate the expected waiting time W_i^∞ for class i as follow.

$$W_i^\infty = \frac{E[Q_i(\infty)]}{\lambda_i} = \frac{1 + \rho^S}{2\lambda(C)(1 - \rho^S)} \quad \text{for all } i = 1, \ldots, n. \tag{23}$$

Notice that the expected waiting time is constant for every i. This result should be intuitively obvious given the service discipline under consideration. In fact, servers are serving always the oldest[‡] customers, thus in essence every customer should spend on average the same amount of time in the system except for over-taking considerations, which are negligible in the heavy traffic regime.

[†]The process $|\xi|$ is known in the queueing literature as a one-sided *regulated Brownian motion* or RBM process (*e.g.* Harrison[13]).

[‡]In terms of arrival time.

6 Conclusions

In this paper we have presented a performance analysis for a multi-class multi-server queueing system with restricted customer-server matchings. These types of systems are common in situations where each class of users can be served only by a specific subset of the available severs. Our analysis of a general n-classes m-servers operation under a First-Come-First-Serve service discipline seems to be novel for this type of system.

Under a Markovian formulation, we first derive stability conditions. These conditions are simple and intuitive and they require that the aggregate arrival rate for any subset B of users' classes has to be strictly less than the service capacity available to serve B, *i.e.*, $\lambda(B) < \mu(S(B))$ for all B.

The central topic of this work, however, has been the characterization of in steady state probability that a class-i arrival is served by server j (P_{ij}). Our interest on these probabilities relates to the underlying assignment occurring in these systems. Depending on the application, a match between a class-i arrival and server j has some specific implications on the operation of the system in terms of benefits and/or costs. For example as policy makers working in the adoption department our goal would be to ensure that infants are adopted by the "right" families or as a manufacturer we would be interested that every job is assigned to the "right" machine. Therefore, in order to address the optimization of the overall system we first need to understand how to compute the P_{ij}'s.

Under heavy traffic conditions, we formalize the mathematical problem as an infinite dimensional matching problem. Unfortunately, we have not been able to solve this problem in closed form. However, borrowing some ideas from the special case of a *completely connected* system we have approximated the value of the P_{ij} with a high degree of precision; on average the error of our computations is below 1% when compared to simulations. Moreover, a simple solution is available which requires the solution of a single $n \times n$ system of linear equations (see equation (16).

We conclude the analysis by computing estimates for the steady state average waiting time and queue length for each class under heavy traffic conditions. As we should expect the average waiting times for all classes coincide, reflecting the fact that any over-taking phenomena are negligible in the heavy traffic limit because of the asymptotic growth of the waiting time with respect to the service time.

We believe our work provides a simple characterization of a multi-class multi-server system with reestricted customer-service matchings under heavy traffic conditions. It has, however, left without answer some important issues that we consider are part of future research. For instance, it would be interesting to generalize our results for moderate traffic intensity under a specific allocation rule ($\mathcal{A}(i, j, \mathcal{S})$). Similarly, the problem of optimizing the operation of the system was not addressed in this work. Specifically given a set of customer classes C, a set of servers S, and a payoff matrix r_{ij} ($i \in C$ and $j \in S$) find the optimal configuration in terms of the matching matrix R that maximizes $\sum P_{ij}\, r_{ij}$. Finally, the quality of our approximations in (16) make us wonder whether this solution is indeed exact or not. At this point we have not been able to provide any proof in either direction leaving this issue as an

open problem.

References

[1] Becker, K.J., D.P. Gaver, K.D. Glazebrook, P.A. Jacobs, S. Lawphongpanich. 2000. Allocation of Tasks to Specialized Processors: A Planning Approach. *Eur. J. Ops. Res.* **126**, 80-88.

[2] Bell, S.L., R.J. Williams. 2001. Dynamic Scheduling of a System with Two Parallel Servers in Heavy Traffic with Resource Pooling: Asymptotic Optimality of a Threshold Policy. *Ann. Appl. Prob.* **11** , 608-649.

[3] Billingsley, P. 1999. *Convergence of Probability Measures*. Wiley Inter-Science.

[4] Borokov, A.A. 1965. Some Limit Theorems in the Theory of Mass Service, II. *Theor. Prob. Appl.* **10**, 375-400.

[5] Dai, J. G. 1995. On positive Harris recurrence of multiclass queueing networks: a unified approach via fluid limit models. *Annals of Applied Probability* **5**, 49-77.

[6] —,G. Weiss. 1996. Stability and Instability of Fluid Models for Re-Entrant Lines. *Math. Ops. Res.* **21** , 115-134.

[7] Filipiak, J. 1984. Dynamic Routing in a Queueing System with a Multiple Service Facility. *Ops. Res.* **32**, 1163-1180.

[8] Foley, R.D., McDonald D.R. (2001). Join the Shortest Queue: Stability and Exact Asymptotics. *Ann. Appl. Prob.*, **11**, 569-607.

[9] Foss, S. 1998. On Stability of a Partially Accessible Multi-Station Queue with State-Dependent Routing. *Queueing Sys.* **29**, 55-73.

[10] Glickman, T. 1975. Resource Allocation to Minimize Delay in a Dual-Purpose Service Facility. *Ops. Res. Quart.* **26**, 305-315.

[11] Green, L. 1985. A Queueing System with General-Use and Limited-Use Servers. *Ops. Res.* **33**, 168-185.

[12] —. 1986. Correction to "A Queueing System with General-Use and Limited-Use Servers". *Ops. Res.* **34**, 184.

[13] Harrison, J.M. 1990. *Brownian Motion and Stochastic Flow Systems*. Krieger.

[14] —. 1998. Heavy Traffic Analysis of a System with Parallel Servers: Asymptotic Optimality of Discrete-Review Policies. *Ann. Appl. Prob.* **8**, 822-848.

[15] Houtum, G.J., I.J.B.F. Adan, J. Wessels, W,H,M. Zijm. 2001. Performance Analysis of Parallel Identical Machines with a Generalized Shortest Queue Arrival Mechanism. *OR Spektrum* **23**, 411-427.

[16] Iglehart, D.L., W. Whitt. 1970. Multiple Channel Queues in Heavy Traffic. *Adv. Appl. Prob.* **2**, 150-177.

[17] Kaplan, E.H. 1984. *Managing the Demand for Public Housing.* ORC Technical Report #183, MIT.

[18] Kaplan, E.H. 1988. A Public Housing Queue with Reneging and Task-Specific Servers. *Decision Sci.* **19**, 383-391.

[19] Roque, D. 1980. A Note on "Queueing Models with Lane Selection. *Ops. Res.* **28**, 419-420.

[20] Schwartz, B.L. 1974. Queueing Models with Lane Selection: A New Class of Problems. *Ops. Res.* **22**, 331-339.

Appendix

Proof of Proposition 2: We prove the result showing that the corresponding fluid limit version of the system is stable so that we can invoke Theorem 4.2 in Dai [5]. The prove shows that in the fluid limit the aggregate queue length process $Q(t) = \sum_{i=1}^{n} Q_i(t)$ has the following property:

$$\text{If } Q(t) > 0 \text{ then } \dot{Q}(t) = \frac{\mathrm{d}Q(t)}{\mathrm{d}t} \leq -\gamma \text{ a.e. for some } \gamma > 0. \tag{24}$$

This Lyapunov form of the aggregate queue length is sufficient for the stability of the fluid model since this condition implies that $Q(t) = 0$ for all $t > Q(0)/\gamma$ (*e.g.*, Lemma 2.2 in Dai and Weiss [6]). The proof of (24) depends essentially in the *work-conserving* nature of our service discipline. In this sense, this proof can be apply to other work-conserving disciplines. Let $t > 0$ be a fixed time and suppose that $Q(t) > 0$, we will consider two cases:

Case 1: Suppose that $Q_i(t) > 0$ for all $i = 1, \ldots, n$. In this case, the work-conserving nature of our service discipline imply that all servers are working and therefore

$$\dot{Q}(t) = \lambda(C) - \mu(S) < 0,$$

where the inequality holds because of (4). Thus, (24) is satisfied in this case.

Case 2: Suppose that $Q(t) > 0$ but there is a proper subset $I \subset C$ of customer classes such that $Q_i(t) = 0$ for all $i \in I$. Let us define $B = C - I$ to be the set of customers classes with positive queue at time t. Given the work-conserving policy under consideration all the server in $S(B) \subseteq S$ are working at their full service rate. However, it is not clear at which rate are working the severs in $S - S(B)$. Figure 9 shows pictorially the situation at time t. Let $\mu(t)$ be total service at which the whole system is working at time t. Clearly in this case $\mu(t) \geq \mu(S(B))$. In addition, servers in the set $S - S(B)$ are working exclusively for customers in the set I. Again, two possibilities arise with respect to the service rate of the servers in I.

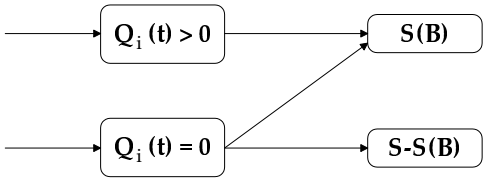


Figure 9: At time t, customers classes are divided in two groups: positive queue (B) and empty queue (I).

(a) Suppose the servers in $S - S(B)$ have enough capacity to server the new arrivals in I. Then, since the queue for I is 0, the server in I will be able to keep the queues in I at the zero level. Thus, the cumulative service rate of the servers in I at time t is exactly $\lambda(I)$ in this case. In this situation we have that

$$\dot{Q}(t) = \lambda(B) - \mu(S(B)) < 0.$$

Again, the last inequality follows directly from (4).

(b) Suppose the servers in $S - S(B)$ do not have enough capacity to server the new arrivals in I. Then, at time t^+ there is a queue growing for some classes in I since the servers in $S(B)$ are giving priority to customers in B. Let $I' \subseteq I$ be the set of customer classes for which a positive queue develops at time t^+. In this situation, we can redefine the set B of classes with positive queue as $B \leftarrow B \cup I'$ and the set of classes with empty queue as $I \leftarrow I - I'$. In this manner, we instantaneously (at t^+) generate new sets B and I such that the situation in Case 2a above is satisfied.

In conclusion, almost everywhere in $t > 0$ the set of empty queues I and the set of positive queues B satisfies case Case 2a above, *i.e.*, the servers in $S - S(B)$ have enough capacity to serve customers in I. From this observation and the results in Case 1 and Case 2a, we conclude that the cumulative queueing process $Q(t)$ satisfies condition (24). Moreover, γ is bounded below by

$$\min_{B \subseteq C} \{\mu(S(B)) - \lambda(B)\},$$

which is guaranteed by (4) to be positive. ∎

Proof of Proposition 5: The prove follows directly from (16) and the recursion in (13)-(14). In fact, from (16) we have that

$$\begin{aligned}
\sum_{i=1}^{n} F_{ij} &= \sum_{i=1}^{n} \widetilde{F}_{ij}^{0} + \sum_{i=1}^{n} M_{ij} V_i - (M'V)_j \sum_{i=1}^{n} L_{ij} \\
&= \sum_{i=1}^{n} \widetilde{F}_{ij}^{0} - (M'V)_j \left(1 - \sum_{i=1}^{n} L_{ij}\right) \\
&= \sum_{i=1}^{n} \widetilde{F}_{ij}^{0} \qquad \text{since } \sum_{i=1}^{n} L_{ij} = 1 \\
&= \mu_j,
\end{aligned}$$

the last equality follows directly from the definition of $\widetilde{F}^0_{ij} = F^0_{ij} - L_{ij}D^0_j$. To prove the second part of the proposition we notice that by the construction of the flows in (13)-(14) we have

$$N^k_i + \sum_{j=1}^{M} F^k_{ij} = \lambda_i.$$

The proof follows directly using induction. In fact, it is straightforward to check the previous condition for $k = 0$. In addition, from (13)-(14) we have that

$$\begin{aligned} N^{k+1}_i + \sum_{j=1}^{M} F^k_{ij} &= (AN^k)_i + \sum_{j=1}^{M} F^k_{ij} + N^k_i \sum_{j=1}^{m} M_{ij} - \sum_{j=1}^{m} L_{ij}(M'N^k)_j \\ &= (A\,N^k)_i + \sum_{j=1}^{M} F^k_{ij} + N^k_i - (A\,N^k)_i \quad \text{since } \sum_{j=1}^{m} M_{ij} = 1 \text{ and } A = L\,M' \\ &= \lambda_i. \end{aligned}$$

The last equality follows from the induction step. ∎