Approximate Analysis
of
an Open Multi-Class
Queueing Network
with
Class-Dependent Population
Constraints

Harry G. Perros
Y. Dallery
G. Pujolle

Center for Communications and Signal Processing
Electrical and Computer Engineering Department
North Carolina State University

CCSP TR-89/20

October 1989
Abstract: We consider a multiclass open queueing network with class-dependent population constraints. That is, the total number of jobs of each class that may be present in the network cannot exceed a given value. A job that arrives at the network during the time that the current number of jobs of the same class is equal to the population constraint, is forced to wait in an external queue. Such queueing networks are especially useful for modelling window-flow control mechanisms. We present a method for obtaining an approximate solution of such a queueing network. The method is based on the use of an equivalent closed queueing network model, which is analyzed using an approximate product-form solution technique. The performance parameters of the original open queueing network are easily derived from the equivalent closed queueing network. Numerical results show that this method is fairly accurate.

Keywords: Window flow control, open queueing networks, multiple classes, population constraints, performance evaluation, approximate analysis.
1. INTRODUCTION

In this paper, we consider an open queueing network with different classes of jobs and population constraints. With the exception of the population constraints, the basic network we consider is of BCMP type [3]. Each class $r$ of jobs has its own population constraint. That is, the total number of class $r$ jobs that can be simultaneously present inside the network can not exceed a fixed number, say $N_r$. Class $r$ jobs that arrive at the network during the time that there are $N_r$ jobs present in the network, are forced to wait in an external queue. As soon as a class $r$ job leaves the network, a job of the same class is allowed to enter the network. Such a queueing network can be used to analyze the performance of sliding window flow control schemes. For instance, in the case of virtual circuit networks, each virtual circuit has its own end-to-end sliding window flow control. That is, the total number of outstanding unacknowledged packets may not exceed a pre-specified number. A similar window flow control scheme can be found in the OSI protocol for connection oriented traffic, where the total number of unacknowledged TPDU (Transport Protocol Data Units) may not exceed a pre-specified number.

Queueing networks with population constraints do not have product form solutions. (Note that the situation we are considering differs from the case where blocked jobs are lost which leads to product form solutions [22]). As a result, several papers have been devoted to the approximate analysis of open queueing networks with population constraints (OQNs-PC). Most of the approximation methods are based on aggregation techniques. The idea behind aggregation is that a subsystem is replaced by a flow equivalent service center (FESC) with load-dependent service rates [7, 24, 34]. The service rates of the equivalent server are obtained by analyzing this subsystem in isolation. Theoretical justification of aggregation techniques is provided in two extreme cases: separable networks [2, 12], and nearly completely decomposable systems [14].

The use of aggregation for the analysis of OQNs-PC was originally proposed by Avi-Itzhak and Heyman [1] for networks with exponential servers and a single class of jobs. The aggregation method is as follows. First, we analyze the network without the external queue as a closed queueing network. Let $X_c(n)$ be its throughput for population $n$, for $n = 1, \ldots, N$, where $N$ is the maximum allowable number of jobs in the open network. The network is then replaced by an FESC with load-dependent service rates $\mu(n)$ given by $\mu(n) = X_c(\min\{n, N\})$, for any $n \geq 1$. Thus, the aggregate system reduces to an $M/M/1$ queue with load-dependent service rates. In the case of an exponential network, the throughput of the closed model can be calculated exactly using any algorithm for separable networks. This aggregation technique may also be used in the case of OQNs-PC with general service time distributions. In that case, the closed model can be analyzed using a method such as Marie's [25]. This aggregation technique has also been used in the case of
queueing networks with nested subnetworks, where each subnetwork has its own population constraints [19].

A direct extension of this aggregation technique to the case of networks with several classes of jobs and class-dependent population constraints was considered by Sauer [33]. The network, without the external queues, is analyzed in isolation as a closed multiclass queueing network for all feasible population vectors. The aggregate system is then replaced by a single queue with service rates dependent on the state vector. This queue is analyzed as a multidimensional Markov chain. This approach can be used for very small models, as it rapidly becomes very complex due to the total number of combinations that are required to be taken into account. In order to avoid the complexity problem introduced by this approach, several techniques have been proposed [6, 23, 26, 36]. In particular, Brandjwan [6] and Lazowska and Zahorjan [23] independently developed the following method. The system is decomposed into a set of independent single-class queues with load-dependent service rates, each queue modelling the behaviour of a particular class of jobs. The load-dependent service rates of a particular class are obtained by analyzing the closed multiclass queueing network for all population values of this class, while the populations of the other classes of jobs are fixed at their average values. An iterative procedure is used to determine the load-dependent service rates of each single-class queue. In addition to the FESC approximation, this method involves an additional approximation which is due to the assumption that the influence of the other classes on a given class can be adequately represented through the use of average values. A simple approximation technique was proposed by Thomassian and Bay [36]. This technique is based on the method of adjusted rate [30, 31]. As in [6, 23], the system is decomposed into a set of single-class FESCs. The load-dependent service rates of the FESC of a given class are obtained as follows. The original multiclass model is analyzed as a single-class model. The influence of the other classes is represented by their contribution to station utilizations. That is, the service rates of this single-class model are obtained by reducing the service rates of the original network. The FESC is then obtained by aggregating the resulting single-class network. This approximation technique is very simple since it does not require the solution of closed multiclass networks. However, as a result of adjusting the service rates, the performance parameters of a given class are calculated independently of the population constraints of the other classes, and this may lead to significant errors.

Multiclass queueing networks with shared population constraint have been considered in [8, 24]. In this case, the total number of jobs (irrespective of their class) that can be simultaneously present in the network may not exceed a pre-specified number. Multiclass queueing networks with population constrained subnetworks have also been studied in [21]. A simpler case was considered in [29].
A different approach for the analysis of OQN-PC has been proposed by Dallery [15]. The model considered in [15] is a single-class open queueing network with general service times. Let N be the maximum allowable number of customers in the network. This capacity limitation can be seen as a limited number of resources. To be allowed to enter the network, a job needs first to get a resource. It will then hold this resource until it leaves the network. The first step of the method is to transform the basic open model into an equivalent closed model. This is done by exchanging the roles of the jobs and the resources. The model obtained is a closed queueing network whose population is equal to the maximum allowable number of customers in the open queueing network. The closed queueing network is composed of the set of stations of the open queueing network plus an additional station which models the external queue. An approximation technique, based on Marie's approximation method [25], is then derived in order to analyze the closed queueing network. The open model can be easily analyzed if we know the solution of the equivalent closed queueing network. It has been shown in [15] that if all service distributions are exponential, then this method provides exactly the same results as the aggregation method. However, in the case of general service times, this method is more accurate than the aggregation method (see [15]). Several extensions of this approach have been considered. In [4], a closed queueing network with general service times and population constrained subnetwork was analyzed. Again it was found that the performance parameters, especially the throughput, were more accurately estimated. In [10], an open multiclass network with general service times and shared population constraints was considered. Using the above method, this network can be equivalently viewed as a closed multiclass network with a single chain. The behaviour of this closed queueing network was approximated by the behaviour of a closed single-class queueing network which was then analyzed using the technique presented in [15].

In this paper, we extend the approach proposed in [15] in order to analyze open multiclass queueing networks with class-dependent constraints. The paper is organized as follows. In section 2, we introduce the open multiclass queueing network with population constraints, and then define the equivalent closed model. In section 3, an approximate solution of the closed model is derived. Section 4 deals with the computational procedure for obtaining performance measures of the closed model and subsequently of the original open queueing network. Numerical results are presented in section 5, and, several extensions are considered in section 6. Finally, an alternative view of this approximation method is presented in the appendix.
2. THE OPEN MODEL AND ITS EQUIVALENT CLOSED MODEL

We consider an open multiclass queueing network with class-dependent population constraints. With the exception of the population constraints, the basic network we consider is of BCMP type [3]. Let M denote the number of service stations and R the number of classes of jobs. Let $q_{0j}(r)$ be the probability that a class r job joins station j upon arrival at the network, and $q_{ij}(r)$ be the probability that a class r job joins station j after service completion at station i. The probability that a class r job leaves the network after service completion at station i, $q_{i0}(r)$, is therefore given by:

$$q_{i0}(r) = 1 - \sum_{j=1}^{M} q_{ij}(r) \quad \text{for } r = 1, \ldots, R \quad (1)$$

The arrival process of class r jobs is Poisson with rate $\lambda_r$. Each station of the network may be of any BCMP type. Especially, any station, say i, with first-come-first-served (FCFS) service discipline has exponentially distributed service times with rate $\mu_i$ independent of the job's class.

In the model considered in this paper, each class of jobs has its own population constraint. Let $N_r$ be the population constraint of class r jobs. That is, $N_r$ is the maximum number of class r jobs that can be simultaneously present inside the network. With each class of jobs is associated an external queue. If an r class job arrives while the current number of class r jobs inside the network is less than $N_r$, it instantaneously enters the network. Otherwise, it has to wait in its external queue. As soon as a class r job leaves the network, the first-in-line job in the class r external queue, if any, enters the system.

The population constraint of each class can be represented by a set of resources, referred to as tokens. Since the population constraints are class-dependent, each class of jobs is associated with a class of tokens. The total number of class r tokens is equal to the population constraint of class r jobs, i.e. $N_r$. Thus, in order to enter the network, a class r job must first get a class r token. The job then holds the token during its sojourn through the network. As soon as the job leaves the network, the token is released and it is immediately available for another job.

This open model is represented in figure 1 using the formalism of queueing networks with synchronization introduced in [18]. The internal BCMP network consisting of the M stations is represented by a circle. Each of these stations will be referred to as an internal station. There is one external queue per class. The external queue together with the token queue will be referred to as the synchronization station. As soon as at least one job and one token are present at a
synchronization station, an assembly takes place and the job/token pair enters the network. When the job leaves the network, the token is fed back into its corresponding external queue of resources.

![Figure 1: The open model with R classes of jobs](image)

It is quite easy now to see how this queueing network can be used to model a set of virtual circuits which share the same communication network. The stations of the queueing network can be seen as representing the nodes of the computer communication network, and a job is identified with a data unit upon which the window-flow control is carried out. The path through the queueing network associated with a class of jobs can be seen as a virtual circuit, and the maximum allowable number of jobs (i.e. the total number of tokens) can be identified with the maximum window size of the virtual circuit. Finally, the synchronization station can be seen as the point where the window-flow control is applied.

An equivalent view of the open queueing network shown in figure 1 can be obtained by exchanging the roles of the jobs and the tokens as proposed in [15]. In particular, consider the tokens as the jobs of the network, and the jobs as resources. The number of classes of jobs is again R since there are R different types of tokens. However, the number of jobs in each class is now a constant. Indeed, the number of class r jobs is equal to the corresponding number of tokens,
that is $N_r$. Therefore, the network is composed of $R$ closed chains of jobs. From this point of view, the system is a closed multiclass queueing network with external resources (the jobs). The closed model consists of the $M$ internal stations of the original network and the $R$ synchronization stations. The number of stations of the closed model is therefore $M+R$. The $M$ stations of the original network are numbered from 1 to $M$ and the $R$ synchronization stations from $M+1$ to $M+R$. Thus, station $M+r$ is the synchronization station corresponding to the $r$th class of jobs. Note that this station is visited only by class $r$ jobs. The population vector is $N = (N_1, N_2, \ldots, N_R)$. This equivalent view is illustrated in Figure 2.

![Figure 2: The equivalent closed model](image)

Let $p_{i,j}(r)$ be the routing probabilities of class $r$ jobs in the closed model. These probabilities can be easily obtained from those of the open model. We have

\begin{align}
    p_{i,j}(r) &= q_{i,j}(r), \quad \text{for } i, j = 1, \ldots, M \\ 
    p_{M+r,j}(r) &= q_{0,j}(r), \quad \text{for } j = 1, \ldots, M \\ 
    p_{i,M+r}(r) &= q_{i,0}(r), \quad \text{for } i = 1, \ldots, M
\end{align}

In the closed model, when a class $r$ job (i.e. a token) arrives at its corresponding synchronization
station (station M+r), it is instantaneously served if at least one external resource (i.e. a job) is available. Otherwise, it has to wait until a resource is available, that is until a class r job arrives at the system. Recall that the arrival process of each class of jobs is Poisson. Each time a job (a class r token) is served at station M+r, one external resource (a class r job) is consumed. Note that in the closed model jobs are consumable resources whereas in the open model tokens are non-consumable resources.

It is important to emphasize that the open and closed models are totally equivalent. They differ from each other only by the roles given to the jobs and tokens. Thus, no approximation is introduced in the transformation of the open model into a closed model. The closed model is just an other way of looking at the same system. Of course, this transformation does not make the problem simpler. In fact, as in the open model, no exact solution of the closed model can be obtained. However, the closed equivalent model is of interest since we can analyze it approximately using known approximation techniques for closed queueing networks, as shown in the next section.

3. APPROXIMATE SOLUTION OF THE CLOSED MODEL

In this section, we extend the technique presented in [15] to derive an approximation solution of the closed model. This technique is based on Marie's approximation method for general single-class queueing networks [25]. The idea of Marie's method is to replace any non-BCMP station, such as a FCFS station with non-exponential service times, by an equivalent exponential server with load dependent service rates. Thus, the original network is approximated by an equivalent product-form network. The load-dependent service rates are obtained by analyzing each station in isolation under a state-dependent Poisson arrival process. An iterative procedure is then used to calculate the unknowns. A slightly different view of Marie's method using operational analysis [17] is presented in [16]. A unified view and a comparison of product-form approximation techniques for single-class closed queueing networks, namely the aggregation technique and Marie's method, is presented in [4].

An extension of Marie's method to queueing networks with different classes of jobs has been proposed in [28]. In general, it is much more complex than in the single-class case. However, in our case, the extension of Marie's method is fairly easy. This is because all the internal stations are BCMP stations. As a result, they do need not to be replaced by an equivalent station. The only non-BCMP stations are the synchronization stations. Now, each of these stations is visited only by a single class of jobs. Therefore, the equivalent station is simply obtained as in the case of single-class networks [15].
The closed model is thus approximated by a multiclass closed queueing network of BCMP type. This equivalent network has M+R stations, the M internal stations (stations 1 to M) and R single-class exponential stations with load-dependent service rates (stations M+1 to M+R). It is illustrated in figure 3. Obviously, the R classes of jobs form a set of R closed chains. Since the equivalent network is of BCMP type, the routing of each class of jobs is characterized only by the average visit rates. Let \( V_i(r) \) be the average visit rate of class r jobs at station i, i.e., one of the solutions of the following system of M+1 equations [3]:

\[
V_i(r) = \sum_{j=1}^{M} V_j(r) \, p_{j,i}(r) + V_{M+r}(r) \, p_{M+r,i}(r) \quad \text{for } i = 1, \ldots, M \text{ and } i = M+r
\]

The solution of (5) is determined up to a multiplicative constant. We choose the solution such that \( V_{M+r}(r) = 1 \). Then, \( V_i(r) \) can be interpreted as the average visit rate of a class r job at station i between two consecutive visits at the synchronization station. Also, it is easy to check that the average visit rates are equal to those of the open model.

Figure 3: The approximate closed queueing network
Let $\mu_{M+r}(n)$ be the load-dependent service rate of the rth equivalent station when n jobs (tokens) are present at this station. Recall that this station corresponds to the synchronization station of class r jobs. We now show how these parameters can be determined. This procedure is similar to that used in the case of single-class networks. Therefore, some of the details will be omitted and the interested reader is referred to [4, 15, 16, 25]. As proposed by Marie [25], the parameters of each equivalent station are obtained by analyzing the corresponding station in isolation fed by a state-dependent Markov arrival process. Let us consider the rth synchronization station, that is station $M+r$. Let $\lambda_{M+r}(n)$ be the state-dependent rate of the Markov arrival process when n jobs (tokens) are present at the station. The model of this synchronization station in isolation is shown in figure 4.

![Figure 4](image.png)

**Figure 4:** Analysis of class r synchronization station in isolation

The steady-state solution can be easily obtained ([15], [19]). Let $p_{M+r}(n)$ be the steady-state probability of having n jobs (tokens) in the class r synchronization station when considered in isolation. Then, the conditional throughputs, $v_{M+r}(n)$, can be obtained as [25]:

$$v_{M+r}(n) = \frac{\lambda_{M+r}(n-1) p_{M+r}(n-1)}{\lambda_{M+r}(n)}$$

for $n = 1, ..., N_r$ \hfill (6)

The conditional throughput is the inverse of the pseudo service time used in [15, 16]. The pseudo service time, $S_{M+r}(n) = 1/v_{M+r}(n)$, is the expected time station $M+r$ spends in state n between two consecutive completions occurring while the state is n. Then, it was shown in [15] that the conditional throughputs are given by:

$$v_{M+r}(1) = \frac{1}{\lambda_{r} \lambda_{M+r}(0) \frac{1}{p_{M+r}(1)}}$$

(7)
\( v_{M+r}(n) = \lambda_r \quad \text{for} \quad n = 2, \ldots, N_r \) (8)

The synchronization station is then replaced by an equivalent station having load-dependent service rates equal to the conditional throughputs in isolation [25], i.e.:

\( \mu_{M+r}(n) = v_{M+r}(n) \quad \text{for} \quad n = 1, \ldots, N_r \) (9)

The analysis of a station in isolation requires the knowledge of the state-dependent arrival rates. Actually, in the case of a synchronization station, only the quantity \( \lambda_{M+r}(0) \) is required. These quantities are obtained from the product-form solution of the equivalent BCMP network [25]. Indeed, provided that the load-dependent service rates of all equivalent stations are known, the resulting BCMP network can be solved exactly. Especially, the marginal probabilities of each equivalent station can be obtained. Let \( p_{M+r}(n) \) be the steady-state probability of having \( n \) jobs at the \( r \)th equivalent station, i.e., station \( M+r \), in the equivalent network. Then, the state-dependent arrival rates are obtained as [16, 25]:

\[ \lambda_{M+r}(n) = \mu_{M+r}(n+1) \frac{p_{M+r}(n+1)}{p_{M+r}(n)} \quad \text{for} \quad n = 0, \ldots, N_r - 1 \] (10)

Note that equation (10) is a reverse form of equation (6). Actually, equations (6), (9), and (10) imply that the marginal probabilities of each equivalent station in isolation and inside the network are equal, i.e., \( p_{M+r}(n) = p_{M+r}(n) \).

As it appears, the load-dependent service rates of the equivalent stations depend on the state-dependent arrival rates (through the analysis of each station in isolation), which in turn depends on the load-dependent service rates (through the analysis of the equivalent network). Therefore, an iterative procedure must be used in order to determine the unknown parameters. This procedure is presented in the next section.

4. ALGORITHMIC SOLUTION

The solution presented in section 3 is obtained by approximating the closed model by an equivalent BCMP network. This network is a multichain closed queueing network consisting of \( M+R \) stations and \( R \) classes of jobs with population vector is \( N = (N_1, N_2, \ldots, N_R) \). The only unknown parameters of this network are the load-dependent service rates of each of the \( R \) single-class equivalent stations. Now, some of these parameters are easily obtained. In particular, from
equations (8) and (9), we have:

\[ \mu_{M+r}(n) = \lambda_r \]

for all \( n = 2, \ldots, N_r \) and \( r = 1, \ldots, R \) \hspace{1cm} (11)

Therefore, the only unknowns are the quantities \( \mu_{M+r}(1) \), for \( r = 1, \ldots, R \). These parameters can be obtained as the solution of the following fixed-point problem. Let us suppose for a moment that we know these quantities. Then, we can solve the closed queueing network using any computational algorithm, (such as the convolution algorithm [9, 11]), in order to obtain the marginal probabilities of each equivalent station, \( p_{M+r}(n) \). Using these probabilities, we can calculate the state-dependent arrival rates \( \lambda_{M+r}(0) \) from equation (10). (Note that only the probabilities \( p_{M+r}(1) \) and \( p_{M+r}(0) \) are needed.) Also, the conditional throughputs \( v_{M+r}(1) \) can be calculated using equation (7). Finally from equation (9), we can obtain a set of new values for the load-dependent service rates which can be used in the next iteration to solve the closed queueing network. We can thus iterate until convergence of the load-dependent service rates. We note that we can directly express the unknowns as a function of the steady-state probabilities of the closed queueing network. That is, using equations (7), (9), and (10), it is easy to show that:

\[ \mu_{M+r}(1) = \lambda_r \left( 1 + \frac{p_{M+r}(0)}{p_{M+r}(1)} \right) \]  \hspace{1cm} (12)

Thus, the following simple iterative procedure can then be used to solve this fixed point problem.

**Computational Algorithm:**

Initialization step:

Set \( \mu_{M+r}(1) = \lambda_r \), for all \( r = 1, \ldots, R \).

Iteration step:

Step 1. Solve the closed queueing network with the current values of \( \mu_{M+r}(1) \), \( r = 1, \ldots, R \).

Step 2. Calculate new values of the load-dependent service rates, \( \mu_{M+r}(1) \), for all \( r = 1, \ldots, R \), using equation (12). Go to step 1 until convergence of the quantities \( \mu_{M+r}(1) \). ♦

Let us now discuss the computational complexity of this approximation technique. Obviously, its computational complexity is related to the solution of the multiclass BCMP network in step 1. If the above algorithm is applied without any particular care, its computational burden may be very high, seeing that the closed queueing network has to be solved at each iteration. Fortunately, this may be circumvented as the closed queueing network obtained at different
iterations differ only from each other by the parameters of the R equivalent stations. This observation can be used to drastically reduce the computational complexity as discussed below.

At each iteration of the algorithm, the closed queueing network, say C, has to be solved using the latest updated values of the quantities $\mu_{M+r}(1)$. Now, let $C_r$ be the closed queueing network obtained from $C$ by removing the $r$th equivalent station, that is station $M+r$. Thus, this network consists of the $M$ internal stations and the $R-1$ synchronization stations corresponding to all classes but class $r$. Let $X_r(N)$ be the throughput of class $r$ jobs in network $C_r$ with population vector $N = (N_1, N_2, ..., N_R)$. Then, using classical formulae of product-form networks [9], it is easy to show that:

$$\frac{p_{M+r}(0)}{p_{M+r}(1)} = \frac{\mu_{M+r}(1)}{X_r(N)}$$

(13)

So, what has to be calculated are the quantities $X_r(N)$, for all $r$. Now, suppose that we use the convolution algorithm [9, 11] to analyze network $C_r$. As it is known, the convolution algorithm calculates the normalizing constants of the network from which the performance parameters, especially the throughputs, can be easily derived. These normalizing constants are calculated recursively starting with one station and step by step incorporating the other stations, one at each step. Each step consists of a convolution operation. Let us now see how this algorithm can be applied to network $C_r$ assuming that the recursion is performed in the order in which the stations are numbered. After $M-1$ steps (convolution operations), we get the normalizing constants corresponding to the internal network, say $C^I$, consisting of the $M$ internal stations. The normalizing constants of network $C_r$ are then obtained by performing $R-1$ additional convolutions corresponding to the synchronization stations. Now, the following points must be emphasized. First, the last $R-1$ convolutions involve only single-class stations. Therefore, the major computational complexity is related to the $M-1$ convolutions required to obtain the normalizing constants of network $C^I$. Secondly, since all networks $C_r$ have the same internal network $C^I$, the normalizing constants of $C^I$ need only be calculated once. Finally, since the internal network $C^I$ remains the same for all iterations, the normalizing constants of $C^I$ need only be calculated once at the first iteration. So, by taking into account these observations, the computational complexity of the algorithm may be significantly reduced. Indeed, it reduces to calculating the normalizing constants of network $C^I$ once, and then at each iteration, performing convolutions with single-class stations.

Thus, the computational complexity of the approximation technique is mainly that of solving a BCMP network consisting of $M$ stations and $R$ closed chains using the convolution
algorithm. This is feasible for networks of moderate size. However, for large networks, this may no longer be feasible since the computational time and space requirements of exact algorithms such as the convolution algorithm grow very rapidly with the size of the network, i.e., number of stations, number of chains, population vector. In such cases, some sort of approximation has to be used to solve the equivalent BCMP network and get the values of \( X^r(N) \) for all \( r \). Many approximation algorithms, mainly based on the MVA algorithm [32] have been proposed for solving BCMP networks (see among others [13, 35, 39]). Now, the \( R \) equivalent stations have load-dependent service rates. Therefore, we need an approximation technique that can handle such service stations. Most of the proposed techniques do not allow stations to have load-dependent service rates. There are however some which do [20, 27, 38]. These techniques can then be used to calculate an approximate value of the throughput \( X^r(N) \) of each network \( C^r \).

It is important to notice that the equivalent network has a particular structure. In particular, the only stations that have load-dependent service rates are single-class stations, i.e., they are visited only by one class of jobs. Moreover, the load dependence has also a special form: all the service rates but the first one are identical, i.e.

\[
\mu_{M+r}(1) \neq \mu_{M+r}(2) = \mu_{M+r}(3) = \ldots = \mu_{M+r}(N_r)
\]  

(14)

These two features may be useful to derive an efficient approximation technique for solving the equivalent network. (This issue was not investigated as it is beyond the scope of this paper.)

Finally, we note that using (13), equation (10) implies:

\[
\lambda_{M+r}(0) = X^r(N)
\]  

(15)

Then, using (7), (9), and (15), the conditional throughput can be expressed as:

\[
\mu_{M+r}(1) = \frac{1}{\frac{1}{\lambda_r} - \frac{1}{X^r(N)}}
\]  

(16)

This expression can be equivalently used in step 2 of the algorithm instead of equation (12), in order to calculate updated values of the conditional throughput \( \mu_{M+r}(1) \).

Once the convergence of the algorithm has been reached, the performance parameters of the closed model can be derived. Especially, the throughput \( X^r_t(r) \) and the mean number of class \( r \)
tokens $Q^T_i(r)$ at station $i$, can be calculated for any internal station as well as for the class $r$ synchronization station. Now, the performance parameters of the open model are easily obtained using the equivalence between the two models. Let $X^I_i(r)$ and $Q^I_i(r)$ denote the throughput and mean number of class $r$ jobs at any internal station $i$. Since in the internal network jobs and tokens are identical, the performance parameters pertaining to jobs in the open model are identical to those pertaining to tokens in the closed model. This holds for all internal stations. Therefore, we have

$$X^I_i(r) = X^T_i(r) = \lambda_r V_i(r) \quad \text{for } i = 1, \ldots, M$$

(17)

Consider now the external queue. Let $X^J_e(r)$ and $Q^J_e(r)$ denote the throughput and mean number of class $r$ jobs at the external queue. From the analysis of the synchronization station, it is easy to show that [15]:

$$X^J_e(r) = X^T_{M+r}(r) = \lambda_r$$

(18)

$$Q^J_e(r) = p_{M+r}(0) \frac{1}{\lambda_{M+r}(0)} - \frac{1}{\lambda_r}$$

(19)

Little's law can then be used to derive the mean time (or mean response time) of each class of jobs at any internal station and at the external station:

$$R^J_i(r) = \frac{Q^J_i(r)}{X^J_i(r)} \quad \text{for } i = 1, \ldots, M$$

(20)

$$R^J_e(r) = \frac{Q^J_e(r)}{\lambda_r}$$

(21)

Finally, the mean system time of a class $r$ job, i.e., the average total time a job spends in the system, is given by:

$$R^J(r) = R^J_e(r) + \sum_{i=1}^{M} V_i(r) R^J_i(r)$$

(22)
The proposed approximation technique for the analysis of the original open model is mainly based on the analysis of its equivalent closed model. A different, yet equivalent view, of this approximation technique is presented in the appendix. The reason for presenting this alternative view is that it can provide a different insight to the proposed technique. Finally, we note that in the case of single-class jobs, the technique presented in this paper reduces to the technique presented in [15]. Furthermore, for single-class BCMP networks, it is equivalent to the aggregation method. In this special case, the algorithm does not require any iterations.

5. NUMERICAL RESULTS

In this section, we discuss the accuracy of the approximation algorithm presented in section 4. The algorithm was implemented on a SUN workstation. The approximation results were then compared to simulation results obtained using the simulation features of QNAP2[37]. Validation results are given for the two queueing networks shown in figure 5 and 11.

The queueing network in figure 5 consists of three stations in tandem and three classes of jobs. Each station is represented by a circle, and it is assumed to be a single server queue with an exponentially distributed service time. For each class, the arrival process of jobs is assumed to be Poisson distributed. The routing of each class is shown by the means of a continuous line.

![Figure 5: Three stations with three classes](image)

This particular configuration was analyzed assuming that a) all three classes are symmetrical, i.e. \( N_1=N_2=N_3=N \), and \( \lambda_1=\lambda_2=\lambda_3=\lambda \) and b) the three classes are not symmetrical, i.e. each class \( r=1,2,3 \), has a different \( N_r \) and a different arrival rate \( \lambda_r \). The performance measure used in all comparisons is the mean waiting time in each station. In figures 6 and 7, we give the approximation and simulation results for the mean time in an external queue, and the mean time in stations 1, 2, and 3, for the symmetrical case. These results are plotted as a function of the utilization \( \rho \) of the external queue, i.e. the percent of time the queue is not empty. Simulation confidence intervals are given only in figure 6, seeing that in figure 7 the approximate and the
simulation results are very close. The results were obtained by varying the arrival rate $\lambda$ from 2.0 to 2.42 while $N=3$ and $\mu_1=8$, $\mu_2=12$, $\mu_3=10$.

The non-symmetrical case of the queueing network shown in figure 5 is reported in figures 8, 9, and 10. In particular, in figure 8, we give the approximate and simulation results for the mean time in the external queue of class 1 as a function of the utilization $\rho_1$ of the external queue. Similar results are given for class 2 in figure 9 as a function of the utilization $\rho_2$ of the class 2 external queue. (The error of the approximate results for the external queue of class 3 is similar to
that of class 2.) In figure 10, we give the approximate and simulation results of the mean time in station 1, 2, and 3 for class 1 as a function of the utilization $\rho_1$ of the class 1 external queue. Simulation confidence intervals are given only in figures 8 and 9, seeing that in figure 10 the approximate and the simulation results are very close. The results were obtained by varying the arrival rates from $(\lambda_1, \lambda_2, \lambda_3) = (1.3, 1.8, 2.3)$, to $(\lambda_1, \lambda_2, \lambda_3) = (1.9, 2.4, 2.9)$. $(N_1, N_2, N_3) = (2, 3, 4)$, and $\mu_1 = 8$, $\mu_2 = 12$, $\mu_3 = 10$.

![Figure 8: Mean time in class 1 external queue vs. $\rho_1$.](image)

Figure 8: Mean time in class 1 external queue vs. $\rho_1$, for the queueing network in figure 1, non-symmetrical case.

![Figure 9: Mean time in class 2 external queue vs. $\rho_2$.](image)

Figure 9: Mean time in class 2 external queue vs. $\rho_2$, for the queueing network in figure 1, non-symmetrical case.
Figure 10: Mean time in station 1, 2, 3 for all classes vs. $\rho_1$, for the queueing network in figure 1, non-symmetrical case.

Similar experiments were carried out for the queueing network shown in figure 11. The queueing network consists of six stations and four classes of jobs. As before, each station is represented by a circle, and it is assumed to be a single server queue with an exponentially distributed service time. For each class, the arrival process of jobs is assumed to be Poisson distributed. The routing of each class is shown by the means of a continuous line. This queueing network was analyzed under a) the symmetrical case, where $N_r = N$ and $\lambda_r = \lambda$ for $r=1,2,3,4$, and b) the non-symmetrical case, where each class $r$, $r=1,2,3,4$, has a different $N_r$ and a different arrival rate $\lambda_r$. For simplicity, it was assumed that stations 1, 2, 5, and 6 have the same service rate. Also, stations 3 and 4 were assumed to have the same service rate.

Figure 11: Six stations with four classes
In figures 12, 13 and 14, we give the approximation and simulation results for the mean time in an external queue, the mean time in station 3 (which is identical to that in station 4), and the mean time in station 1 (which is identical to that in stations 2, 5, 6), for the symmetrical case. These results are plotted as a function of the utilization of the external queue. The results were obtained by varying the arrival rate from 4.5 to 5.8, while $N=5$, and $\mu_1=\mu_2=\mu_5=\mu_6=14$, $\mu_3=\mu_4=30$.

![Figure 12: Mean time in external queue vs. $\rho$, for the queueing network in figure 7, symmetrical case.](image1)

![Figure 13: Mean time in station 3 vs. $\rho$, for the queueing network in figure 11, symmetrical case.](image2)
The non-symmetrical case of the queueing network shown in figure 11 is reported in figures 15, 16, and 17. In figure 15, we give the approximate and simulation results for the mean time in the external queue of class 1 as a function of the utilization $\rho_1$ of the external queue. Similar results are given in figure 16 for class 3, as a function of the utilization $\rho_3$ of the class 3 external queue. (The error of the approximate results for the external queues of class 2 and class 4 is similar to that for class 3.) In figure 17, we give the approximate and simulation results of the mean time in station 1, 3, and 6 for class 1 as a function of the utilization $\rho_1$ of the external queue of class 1. These results are representative of the type of results obtained for each class. Simulation confidence intervals are given only in figures 15 and 16, seeing that in figure 17 the approximate and the simulation results are very close. The results were obtained by varying the arrival rates from $(\lambda_1, \lambda_2, \lambda_3, \lambda_4) = (5, 4, 3, 2)$, to $(\lambda_1, \lambda_2, \lambda_3, \lambda_4) = (6.5, 5.5, 4.5, 3.5)$. $(N_1, N_2, N_3, N_4) = (5, 4, 3, 2)$ and $\mu_1 = \mu_2 = \mu_5 = \mu_6 = 14$, $\mu_3 = \mu_4 = 30$. 

Figure 14: Mean time in station 1 vs. $\rho$, for the queueing network in figure 11, symmetrical case.
Figure 15: Mean time in class 1 external queue vs. $\rho_1$, for the queueing network in figure 11, non-symmetrical case

Figure 16: Mean time in class 3 external queue vs. $\rho_3$, for the queueing network in figure 11, non-symmetrical case
In general, it appears that the performance parameters of the internal stations are always very accurately estimated (see figures 7, 10, 13, 14, and 17). The accuracy of the results corresponding to the external queues depends on the load of the system. A fairly good measure of the load of the system is provided by the percentage of time each external queue is non-empty, referred to as the utilization of the external queue. This utilization increases when either the arrival rate increases or the window size decreases. Let us first consider the case where all classes are symmetrical. For these cases, when the load increases (as a result of the increase of the arrival rates), the error increases (see figures 6 and 12). Similar observations were made in the case of single-class networks. Now, in the non-symmetrical cases, the difference in the accuracy of the results pertaining to different classes may be significant (compare figures 8 and 9, and 15 and 16). In general, it seems that for reasonable loads of the system, the proposed method has a satisfactory accuracy. Its accuracy is comparable to the one obtained in the case of single-class networks.

On all the examples we tested, the algorithm always converged. The convergence can actually formally be proved in the special case of two classes of jobs. The number of iterations required to achieve a reasonable accuracy is of the order of 10, and it increases as the load of the system increases.
6. EXTENSIONS

In this section, we briefly discuss several extensions of the method presented in this paper. These extensions are discussed independently of one another, but they can easily be combined.

A first extension is related to the feedback of tokens. So far, it was assumed that a token is immediately available upon departure of a job. This may not be the case in real systems. The algorithm can be easily extended to allow a non-zero delay for the return of tokens. For instance, tokens may go back following the reverse path of the jobs, or following a different route. With this additional assumption, the equivalence between the open and closed models still holds, but the average visit rates of tokens differ from those of jobs. In particular, let $V^T_i(r)$ be the average visit rate of class $r$ tokens at station $i$, and let $V^J_i(r)$ be the average visit rate of class $r$ jobs at station $i$. It can be easily shown that: $V^T_i(r) \geq V^J_i(r)$. The closed model can then be approximately analyzed using the method presented in section 4. Note that since the jobs of the closed model are the tokens, the average visit rates of the closed model are those corresponding to tokens, i.e. $V^T_i(r)$. Let $Q^T_i(r)$ and $R^T_i(r)$ denote the mean queue length and mean response time of class $r$ tokens at station $i$ in the closed model. Then the performance parameters of the open model can again be derived. Let $X^I_i(r)$, $Q^I_i(r)$, and $R^I_i(r)$ denote the throughput, mean queue length, and mean response time, of class $r$ jobs at station $i$, respectively. Then, we have

$$X^I_i(r) = \lambda_r V^I_i(r) \quad \text{for } i = 1, ..., M$$

$$Q^I_i(r) = \frac{V^J_i(r)}{V^T_i(r)} Q^T_i(r) \quad \text{for } i = 1, ..., M$$

$$R^I_i(r) = R^T_i(r) \quad \text{for } i = 1, ..., M$$

Finally, the mean system time of a class $r$ job, $R^J(r)$, is

$$R^J(r) = R^J_e(r) + \sum_{i=1}^{M} V^J_i(r) R^J_i(r)$$

A second extension that can be handled by the approximation algorithm, is to allow the external arrival process of any class to be state-dependent. Let $\lambda_r(.)$ be the rate at which class $r$ jobs arrive. Two cases may be considered. First, this rate may depend on the total number of class $r$ jobs, $n_r$, currently present in the system, i.e. either inside the network or waiting in the external
queue. This type of state-dependency can be used to model a finite source of jobs. In that case \( \lambda_r(n_r) = 0 \) for \( n_r \geq K_r \), where \( K_r \) is the population of class \( r \) jobs. Secondly, this rate may depend only on the total number of jobs currently present in the external queue, \( m_r \). This can be used to model, for instance, a finite buffer capacity of the external queue. In that case \( \lambda_r(m_r) = 0 \) for \( m_r \geq B_r \), where \( B_r \) is the capacity of class \( r \) external queue. That is, a job arriving while the external queue is full is lost. Incorporating the above two types of state-dependent arrivals in our approximation method is very easy, as it only needs to slightly modify the analysis of the individual synchronization stations. (For details see [4].)

A third extension is to incorporate open classes of jobs without population constraints as well as closed classes of jobs. That is, in addition to the \( R \) open classes with population constraints, there are also \( R_O \) open and \( R_C \) closed classes of jobs. First, the effect of the \( R_O \) open classes of jobs can be taken into account by reducing the service rates of the stations using the method of adjusted rate [30, 31]. Although this transformation is not exact since the original network is not BCMP (due to the population constraints on the other \( R \) open classes), it usually provides very good results. Now, we end up with a network with \( R \) open classes with population constraints and \( R_C \) closed classes. In a similar way as we did in section 2, we can define an equivalent closed model. This model consists of \( R + R_C \) closed chains of jobs: the \( R \) classes of tokens and the \( R_C \) classes of jobs. An approximate solution of this closed model can be obtained with the method presented in sections 3 and 4. The only difference is that the equivalent network has \( R_C \) extra classes of jobs. Note that there are still only \( R \) equivalent stations.

A fourth extension is to allow some classes of jobs to share the same population constraint. This may be obtained by combining the treatment of open multiclass queueing networks with shared constraints as proposed in [10] with the method presented in this paper. The idea behind this extension can be easily presented by the means of an example. Let us consider a system with 4 classes of jobs. Classes 1 and 2 share the same population constraints, \( N_{1-2} \). That is, the total number of class 1 and class 2 jobs inside the network can not exceed \( N_{1-2} \). A class 1 (or class 2) job arriving in the system while there are already \( N_{1-2} \) jobs of class 1 or 2 is forced to wait in an external queue. This queue contains all jobs of classes 1 and 2 waiting to enter the network. Then, when a job of either class 1 or class 2 leaves the network, the first job in the external queue is allowed to enter the network. Similarly, we assume that classes 3 and 4 share the same population constraint, \( N_{3-4} \). The open model can again be transformed into an equivalent closed model. However, this closed model is slightly more complicated as it consists of 4 classes of jobs but only 2 closed chains. Chain 1-2 (respectively 3-4) has a constant population \( N_{1-2} \) (respectively \( N_{3-4} \)). Within each chain, switching between classes occur at the synchronization station. A token of chain 1-2 may switch class according to the class of the job it is associated with. This closed
model can then be transformed into a closed model with two chains and two classes of jobs. Again, although this transformation (aggregation of classes within each chain) is not exact since the closed network is not BCMP (due to the synchronization stations), it should only introduce a moderate error. The resulting closed network can again be analyzed with the method proposed in section 3 and 4. The performance parameters of the original open network can then be derived. (For details see [10].)

Finally, we mention that it may be possible to incorporate non BCMP stations in the original network, for instance FCFS exponential stations with class-dependent service times or stations with priorities. This is possible as long as we can obtain an approximate solution of the equivalent network (see section 3).
REFERENCES


[18] S. Fdida, G. Pujolle, and D. Mailles, Réseaux de files d'attente avec sémaphores, *TSI* 5,


[35] P. Schweitzer, Approximate analysis of multiclass closed networks of queues, Int. Conf. on
Stochastic Control and Optimization, Amsterdam, 1979.


APPENDIX

In this appendix, we briefly present an alternative view of the approximation technique proposed in this paper. Consider again the open model defined in section 2 and illustrated in figure 1, and let us consider the input process of class r jobs to the internal network. This input process is also the output process of the external queue of class r jobs. Because of the population constraint (at most \( N_r \) jobs of class r can be inside the network), this process is no longer Poisson (in fact it is not a renewal process either). Obviously, this process is dependent on the state of the internal network. So, the idea is to approximate this process by a state-dependent Poisson process.

Let \( \lambda_r^*(k_r) \) be the arrival rate of this process given that there are \( k_r \) jobs of class r inside the network. Let us now characterize these rates. When there are already \( N_r \) class r jobs inside the network, this input process is interrupted. Therefore,

\[
\lambda_r^*(N_r) = 0
\]  

(A1)

On the other hand, when there are strictly less than \( N_r - 1 \) class r jobs inside the network, there are at least two tokens available and therefore, a new class r job arriving from outside (according to the Poisson process with rate \( \lambda_r \)) will instantaneously be allowed to enter the network. As a result, we have:

\[
\lambda_r^*(k_r) = \lambda_r \quad \text{for all } k_r < N_r - 1
\]  

(A2)

Now, the case where \( k_r = N_r - 1 \), i.e., there is exactly one token available, is more involved. In this case, the rate depends on whether the external queue of jobs is empty or not at the instant of transition into state \( k_r = N_r - 1 \). If it is empty, then the rate is equal to the arrival rate of jobs, i.e. \( \lambda_r \). Otherwise, the rate is infinite since there is at least one job and one token available. So, the rate \( \lambda_r^*(N_r - 1) \) is a weighted average of these two rates. Especially, it is such that : \( \lambda_r < \lambda_r^*(N_r - 1) < \infty \). In order to determine this rate, it is convenient to use the following conservation of flow equation:

\[
\sum_{k_r = 0}^{N_r} p_r^*(k_r) \lambda_r^*(k_r) = \lambda_r
\]  

(A3)
where $p^*_r(k_r)$ is the steady-state marginal probability of having $k_r$ jobs inside the network. The left hand side of equation (A3) is the average input rate to the inside network. Provided that the system is stable, it is equal to the arrival rate of jobs into the system, i.e. $\lambda_r$. Now, using (A1), (A2), and (A3), after some manipulation we get:

$$\lambda^*_r(N_r-1) = \lambda_r \left(1 + \frac{p^*_r(N_r)}{p^*_r(N_r-1)}\right) \quad \text{(A4)}$$

Following this approach, the open model can be approximated by an open queueing network which is again of the BCMP type. It consists of $M$ internal stations and $R$ open chains with state-dependent arrival rates. The unknown parameters are now the quantities $\lambda^*_r(N_r-1)$ for all $r$. Since they depend on the probabilities $p^*_r(k_r)$ which in turn depend on these parameters, they can be determined using an iterative procedure.

We now show that this approach is equivalent to that presented in the main part of this paper. It is well known that the above open queueing network with state-dependent arrivals can be transformed into an equivalent closed network by modelling each external arrival process by an additional station. The resulting closed network consists of the $M$ internal stations plus the $R$ external single-class stations modeling the arrival of jobs. It has $R$ closed chains. Again, let $M+r$ denote the index of the $r$th external station. Let $\mu^*_M(n)$ be its load-dependent service rate. Then, we have

$$\mu^*_M(n) = \lambda^*_M(N_r-n) \quad \text{for } n = 1, \ldots, N_r \quad \text{(A5)}$$

Thus, we end up with a closed queueing network that has the same structure as the one considered in section 3. In order to show that these two networks are the same, it suffices to show that the service rates of the equivalent stations are defined in the same way. Let $p^*_M(n)$ be the steady state probability of the $r$th external station. Then

$$p^*_M(n) = p^*_r(N_r-n) \quad \text{for } n = 1, \ldots, N_r \quad \text{(A6)}$$

As a result, using (A2), (A4), (A5), and (A6), the service rates of the $r$th external station can be written as follows
\[ \mu_{M+r}^*(n) = \lambda_r \quad \text{for } n = 2, \ldots, N_r \quad (A7) \]

\[ \mu_{M+r}^*(1) = \lambda_r \left( 1 + \frac{p_{M+r}^*(0)}{p_{M+r}^*(1)} \right) \quad (A8) \]

Now, these service rates are exactly the same as those obtained in section 4 seeing that (A7) and (A8) correspond respectively to (11) and (12).