A Survey of Closed Queueing Networks with Finite Buffers

Raif O. Onvural

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

June 1987

CCSP-TR-87/7

Abstract

Closed queueing networks with finite buffers are used to model systems such as production systems, communication systems, computer systems and flexible manufacturing systems. Comprehensive study of the results reported in recent years would benefit various research communities as well as practitioners. In this paper, it is attempted to give a systematic presentation of the literature related to such queueing networks.

I. Introduction

System performance has become a major issue in the design and implementation of systems such as computer systems, production systems, communication systems and flexible manufacturing systems. The success or failure of a system often is judged on the degree to which performance objectives are met. Thus, tools and techniques for predicting performance measures are of great interest.

Queueing network models are the most widely used analytical method for estimating the standard performance measures of such systems. Gordon and Newell [24] showed that solution of a closed queueing network, under some restrictive assumptions, has the product form. Baskett, Chandy, Muntz and Palacios [13] extended the range of the product form networks relaxing some of those assumptions. Efficient algorithms, such as Mean Value Analysis [15] and Buzen's algorithm [16], have been developed in the literature for the calculation of performance measures of closed queueing networks with product form solutions.

In recent years, there has been a growing interest in the development of computational methods for the analysis of queueing networks with finite buffers. This is primarily due to a growing need to model actual systems which have finite capacity resources. An important feature of system with finite buffers is that a server may become blocked when the capacity limitation of the destination queue is reached. Various blocking mechanisms have been considered in the literature so far. These blocking mechanisms arose out of various studies of real life systems. For example, in store and forward communication networks, a node can not transmit a message until the destination node has a buffer available. In I/O subsystems, a disk is blocked because the channel is busy transferring data for another disk. Likewise, in an assembly line, a unit finishing its service at a station can not leave because there is no space available in the destination station.

Queueing networks with finite buffers are in general difficult to treat. Such networks could not be shown to have product form solutions. However, certain closed queueing networks with finite buffers have been reported in the literature as having product form solutions.

This paper gives a survey of exact, approximate and numerical results related to closed queueing networks with finite buffers. The most commonly used blocking mechanisms that exist in the literature are defined next. Section III deals with two-node closed queueing networks. In sections IV to VII, we survey results related to different types of blocking mechanisms that are defined in Section II. The concept of indistinguishable nodes is introduced in Section VII and it is shown that symmetrical networks can be solved exactly on a reduced state space.

II. Blocking Mechanisms

Various blocking mechanisms have been reported in the literature so far. These distinct types of models for blocking arose out of various studies of real life systems. Onvural and Perros [29] have classified the most commonly used blocking mechanisms as follows:

TYPE 1:A customer upon completion of its service at queue i attempts to enter destination queue j. If queue j at that moment is full, the customer is forced to wait in front of server i until it enters destination queue j. The server remains blocked for this period of time and it can not serve any other customer waiting in its queue. This blocking mechanism has been used to model systems such as production systems and disk I/O subsystems (cf. Altiok and Perros[8,9],Perros[33]).

TYPE 2:A customer in queue i declares its destination queue j before it starts its service. If queue j is full, the ith server becomes blocked. When a departure occurs from destination queue j, the ith server becomes unblocked and the customer begins receiving service. This blocking mechanism has been used to model systems such as production systems and telecommunication systems. (cf. Boxma and Konheim [14] ,Gershwin and Berman [22]).

Depending upon whether the customer is allowed to occupy the position in front of the server when the server is blocked, we distinguish the following two sub- categories.

TYPE 2.1:Position in front of the server can not be occupied when the server is blocked.

TYPE 2.2:Position in front of the server can be occupied when the server is blocked.

TYPE 3:A customer upon service completion at queue i attempts to join destination queue j. If queue j at that moment is full, the customer receives another service at queue i. This is repeated until the customer completes a service at queue i at a moment that the destination node is not full.

Within this category of blocking mechanisms, we distinguish the following two sub-categories.

TYPE 3.1:Once the customer's destination is determined it can not be altered. This blocking mechanism arose in modelling telecommunication systems (cf. Caseau and Pujolle [17]).

TYPE 3.2:A destination node is chosen at each service completion independently of the destination node chosen the previous time. This type of blocking is associated with reversible queueing networks with blocking (cf. Yao and Buzacott [40,41,42,43]).

We note that in the above mentioned blocking mechanisms a server becomes unblocked when the number of customers in the destination node drops below its maximum capacity. Latouche and Neuts[27] considered other extensions whereby unblocking of a server occurs when the number of units in the destination node drops below a predefined level, not necessarily equal to its maximum capacity. Such blocking mechanisms are not considered here.

Comparisons between these distinct types of blocking mechanisms carried out by Caseau and Pujolle [17], Altiok and Stidham [7], Bocharov and Albores [10], Balsamo, Persone and Iazeolla [11], Onvural and Perros [29] and Onvural [32]. The objective of these comparisons was to obtain an equivalency between different blocking mechanisms applied to the same queueing network. Two blocking mechanisms are said to be equivalent if the network under consideration has the same rate matrix under both types of blocking mechanisms. We note that, all of the equivalencies obtained in the literature assume service time distributions to be exponential. Furthermore, these equivalencies, unless mentioned otherwise, are true only for the specific topologies shown in Figures 1 and 2.

Lemma 1: Types 2.2 and 3.1 are equivalent independent of the topology.

Lemma 2: In cyclic networks (Figure 1):

a)Types 3.1 and 3.2 are equivalent

b)Type 2.1 given buffer capacities B_i is equivalent to type 1 with buffer capacities $B_i - 1$, i=1,...,N; where N is the number of nodes.

Lemma 3:In the central server model (Figure 2):

a)Types 2.1 and 2.2 are equivalent if $B_1=\infty$

b)Types 3.1 and 3.2 are equivalent if $B_i=\infty$, i=2,...,N

c)Types 2.1 and 2.2 are equivalent if $B_i=\infty$, i=2,...,N

Furthermore, there always exists an equivalency between the blocking mechanisms defined above if the network has two nodes with exponential servers.

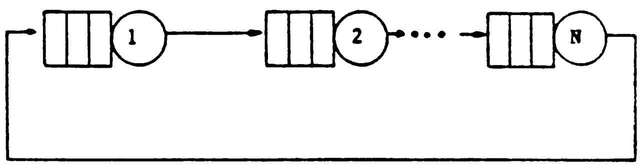


Figure 1:Cyclic Network

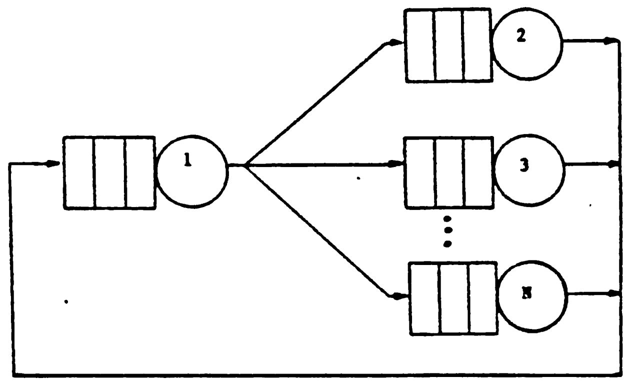


Figure 2:The Central Server Model

III. Two Node Closed Queueing Networks with Finite Buffers

We will start our review of literature with two-node closed queueing networks (cf. Perros [33]) as shown in Figure 3.

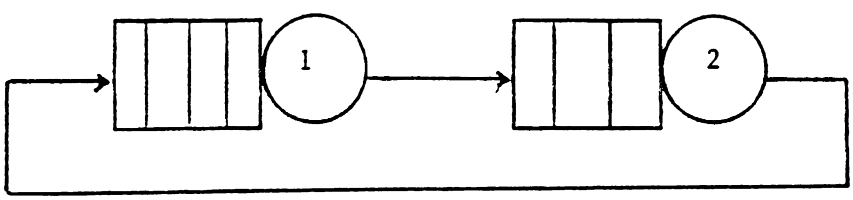


Figure 3: A two − node closed queueing network

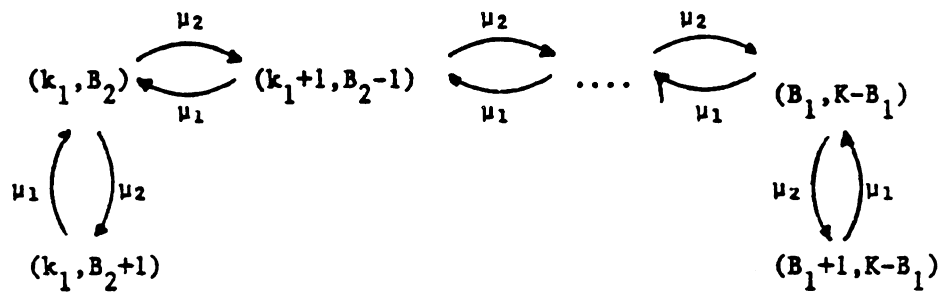
Let K be the number of customers in the network and B_i be the capacity of node i including the space in front of the server. Furthermore, assume that service time at each node is distributed exponentially with rate μ_i, i=1,2.

III.I. Type 1 Blocking Mechanism

Two-node closed queueing networks under type 1 blocking have been studied by Diehl [21] and Akyildiz [1]. In particular, Akyildiz [1] demonstrated that a two-node closed queueing network with finite buffers and with K customers is equivalent to a non-blocking network (i.e. with infinite buffer capacities) with K' customers, where

$$K' = min(N, B_1 + 1) + min(N, B_2 + 1) - N \tag{1}$$

To summarize the equivalency, define k_1=max(0,K-B_2) and k_2=min(K,B_1). Then, the rate diagram associated with two node network has the following simple form:



where, k_1^b, $(K-k_2)^b$ denotes that nodes 1 and 2 are blocked respectively. The equivalent non-blocking network will have the same rate diagram with k_1^b=0, B_1=B_2= k_2=K'. From the above rate diagram, we have:

$$p(k_1 + i, B_2 - i) = (\mu_1/\mu_2)^{i+1} p(k_1^b, B_2) \tag{2}$$

and, $p(k_1^b, B_2)$ can be determined from the normalizing equation, i.e.

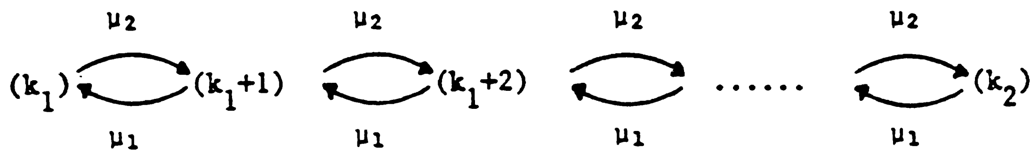
$$\sum_{(i,j)\ feas} p(i,j) = 1 \tag{3}$$

where, p(i,j) is the steady state joint queue length distribution of the two-node network under consideration.

Finally, we note that, these results are readily applicable to a two-node closed queueing network under type 2.1 blocking mechanism after the buffer capacities are increased by one.

III.II. Type 2.2 Blocking Mechanism

Now, consider the two-node closed queueing network under type 2.2 blocking mechanism. This model was first studied by Gordon and Newell[23]. Define, $k_1 = \max(0, K - B_2)$ and $k_2 = \min(K, B_2)$. The rate diagram associated with the network has the following form:



Hence, one can easily obtain that:

$$p(k_1 + i, K - k_1 - i) = (\mu_1/\mu_2)^i \, p(k_1, K - k_1) \tag{4}$$

and, $p(k_1, K - k_1)$ can easily be obtained from the normalizing equation:

$$\sum_{(i,j)\ feas} p(i,j) = 1 \tag{5}$$

Note that, type 2.2 blocking mechanism is equivalent to types 3.1 and 3.2 blocking mechanisms in two-node closed queueing networks. Thus, solution of the two-node network presented above for type 2.2 blocking is also applicable to both types of blocking mechanisms.

We now proceed with the survey of closed queueing networks with finite buffers with more than two nodes. Consider closed queueing networks consisting of N nodes and K customers. B_i is the capacity of node i including the service space in front of the server. A customer upon completion of its service at node i attempts to enter node j with probability p_{ij}, i=1,...,N; j=1,...,N. Parameters of the networks under consideration are summarized in Table 1.

-K:# of customers in the network
-N:# of nodes
-p_{ij}:fraction of departures from node i that proceeds next to station j
-All nodes have FCFS service discipline
-Each node has a single server
-B_i:Capacity of node i

Table 1:Parameters of closed queueing network

IV. Type 1 Blocking

Earlier work in queueing networks with finite buffers was motivated by open queueing networks under type 1 blocking mechanism (also called classical blocking). However, there are relatively few results reported in the context of closed queueing networks and much of the work has been done in recent years (cf. Akyildiz [1,2,5,6], Suri and Diehl [39], Onvural and Perros [30,31], Onvural [32], Diehl [21], and Persone and Grillo [20]).

Now, consider a closed queueing network with parameters as given in Table 1. Furthermore, assume that service time at each node is distributed exponentially with rate μ_i, i=1,...,N. For $1 \leq K \leq \min B_i$, there is no blocking and the network has a product form solution (cf. Gordon and Newell [24]). When $K \geq \min B_i+1$ blocking occurs. In this case, product form solutions are, in general, not available. However, when $K=\min B_i+1$, there can be at most one node blocked at a time. Hence, while a server is blocked, there can not be any customer waiting in its queue. Thus, during the blocking period, the server space in front of the blocked server behaves like an additional buffer space for the blocking node. The following lemma, proved in Onvural and Perros [30], explains this phenomenon.

Lemma 4: Let us consider a closed exponential queueing network with parameters as given in Table 1. If the number of customers in the network, K, is equal to minimum buffer capacity plus one, then the network has a product form solution.

Let, $p(i_1,i_2, \ldots, i_N)$ and $\pi(i_1,i_2, \ldots, i_N)$ be the steady state queue length distribution of a closed queueing network with blocking and with infinite buffers respectively. Note that, $\pi(i_1,i_2, \ldots, i_N)$ has a product form solution. If the assumptions of Lemma 4 are satisfied, then:

$$p(i_1,..,i_N) = \begin{cases} \pi(i_1,..,i_N) & \text{if } \textit{no node is blocked} \\ \dfrac{p_{lj}\, e_l}{e_j} \pi(0,..,i_j = B_j + 1,0,..,0) & \text{if } \textit{node } l \textit{ is blocked by node } j \end{cases} \tag{6}$$

where e_i is the mean number of visits a customer makes to ith node and is given by:

$$e_i = \sum_{j=1}^{N} e_j p_{ji},\ i = 1,\ldots,N \tag{7}$$

with $e_j = 1$ for some j.

IV.I. Throughput

Throughput of a node is defined as the rate at which customers depart from that node. Let $\lambda_i(K)$ and $\lambda(K)$ be throughput of node i and throughput of the network with K customers respectively. Then, we have: $\lambda_i(K) = \{1 - P_i^K(0) - P_i^K(b)\}\mu_i$ where $P_i^K(0)$ and $P_i^K(b)$ are the probabilities that node i is empty and blocked respectively given that there are K customers in the network. $1/\mu_i$ is the mean service time at node i. Furthermore,

$\lambda_i(K) = \lambda(K)e_i$, $i = 1,\ldots,N$, where e_i is given by equation 7.

Clearly, $\lambda(K)$ depends on the parameters of the network. In figure 4, we give an example of $\lambda(K)$ as K changes from 1 to the capacity of the cyclic network shown in figure 1 with $N=3$. Note that, in figure 4, $\lambda(K)$ increases as K increases until it reaches a maximum value, λ^*, for some K^*. For $K > K^*$, $\lambda(K)$ is non-increasing. For the following lemmas, Onvural and Perros [31] used the following two properties without any proof. However, the validity of these assumptions can be shown by stochastical ordering (cf. Shantikumar [37]) at least for some range of values of K.

Property 1: $P_i^K(0)$ is non-increasing as K increases until it reaches zero.

Property 2: For the range of values that $P_i^K(b)$ is non-zero, $P_i^K(b)$ is non-decreasing as K increases.

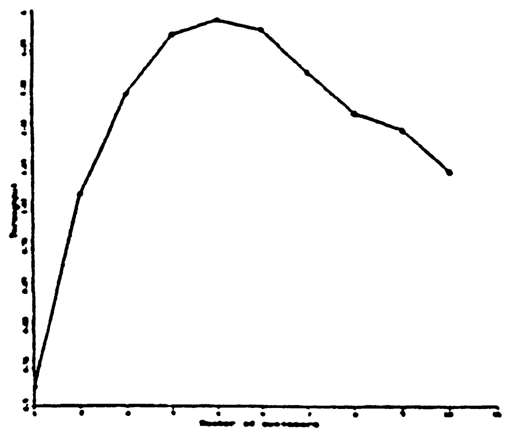


Figure 4: Throughput vs # of customers

Lemma 5: Consider a closed exponential queueing network with parameters as given in table 1, and let $\lambda^* = \max_K\{\lambda(K)\}$, $n = \min_{i=1,\dots,N}\{B_i\}$, $M = \sum_{i=1}^{N} B_i$, and $\beta(K)$ be the throughput of a closed queueing network with same parameters and infinite buffer capacities. Then;

$$\beta(n+1) \le \lambda^* \le \beta(M - \max_{i=1,\dots,N}\{B_i\}+1)$$

Lemma 6: Consider the network defined in Lemma 5. Let K^* be such that $\lambda^* = \lambda(K^*)$. Then;

$$\max_{i=1,\dots,N}\{\min_{j=1,\dots,N}\{B_j \text{ s.t. } p_{ij} \neq 0\}\} \le K^* \le \sum_{i=1}^{N} B_i - \max_{i=1,\dots,N}\{B_i\}+1$$

The above two lemmas provide bounds on the maximum throughput (w.r.t. K) and the number of customers that produces the maximum throughput respectively. However, we should note that these bounds are not usually tight.

Akyildiz [2,6], developed approximation algorithms for the throughput of closed queueing networks with exponential and general service times. He finds an equivalent non-blocking network which has the same or almost equal number of states as the blocking network in which case, both systems have the same behavior and the throughput of both systems are almost equal. The only assumption in the algorithm is that the network under consideration should be deadlock-free.

Due to the blocking mechanism under consideration, and due to the fact that nodes of the network are arbitrarily interconnected, it is possible that deadlocks may occur. For instance, assume that node i is blocked by node j. Now, it is possible that a customer at node j may, upon completion of its service, choose to go to node i. If node i is , at that time, full then deadlock will occur. Usually, in networks with deadlocks, it is assumed that deadlocks are detected immediately and resolved by instantaneously exchanging blocking units. The following lemma gives the sufficient and necessary condition for a network to be deadlock free. (cf. Akyildiz [3]).

Lemma 7:A closed queueing network with finite buffer capacities is deadlock free if and only if for each cycle, C, in the network, the following condition holds:

$$K < \sum_{j \in C} B_j$$

Simply stated, the total number of customers in the network must be smaller than the sum of buffer capacities in each cycles.

Akyildiz's algorithm for closed exponential closed queueing networks can be summarized as follows:

S0: For a given deadlock free exponential network with parameters as given in table 1, calculate the number of states of the blocking network, Z', as follows:

$$Z' = Z_1 * Z_2 * \ldots * Z_N$$

where * is a convolution operator, and Z_i, i=1,...,N is a k+1 dimensional vector given by:

$$Z_i = (z_0^i, z_1^i, \ldots, z_K^i) \quad where \quad z_j^i = \begin{cases} 1 & \text{for } 0 \le j \le B_i + 1 \\ 0 & otherwise \end{cases}$$

S1: Find l such that $\left\{ \begin{bmatrix} N+l-1 \\ N-1 \end{bmatrix} - Z' \right\}$ is minimum. Then, $\lambda(K) = \beta(l)$.

Note that $\begin{bmatrix} N+l-1 \\ N-1 \end{bmatrix}$ is the number of states in a closed queueing network with l customers and $\beta(l)$ is the throughput of the network with infinite buffer capacities and l customers. Hence, the algorithm finds a product form network which has approximately the same number of states as the blocking network while all parameters other than the number of customers in the network are kept the same. The algorithm is very simple to implement and quite accurate. Furthermore, Akyildiz [6] applied the same algorithm to closed queueing networks with finite buffers and with general service times. In addition to the assumptions of table 1, assume that the service time at each node is general with coefficient of variation c_i^2 and having rational laplace transform. The steps of the algorithm are given below:

S0: Given a deadlock free closed queueing network with parameters given in table 1, generally distributed service times with mean $1/\mu_i$ and coefficient of variation c_i^2, find a non-blocking network with K' customers.

S1: Solve the non-blocking network with general service times approximately using Marie's method [28] and calculate the throughput, $\beta(K')$, of the network. Then, $\lambda(K)=\beta(K')$

Finally, we should note that approximately equivalent behavior of the blocking network to the product form network assumed in the above two algorithms is valid only for the throughput. The procedure is not applicable for the calculation of other performance measures.

Another approximation algorithm for the mean response time (equivalently for the throughput) is developed by Suri and Diehl [39]. Their algorithm is applicable to both type 1 and type 2.2 blocking mechanisms, and it will be presented in Section V.

Onvural and Perros [31] developed an approximation algorithm to calculate the throughput of large closed exponential queueing networks with finite buffers. The main steps of the algorithm is given below:

S0: Find K^* (approximately) such that $\lambda(K^*)\geq\lambda(K)$, $K=1,\ldots,\sum_{i=1}^{N} B_i$. Solve the blocking network numerically with K^* customers to calculate $\lambda(K^*)$

S1: Calculate $\lambda(1),\ldots,\lambda(\min B_i+1)$ using one of the efficient algorithms for product form networks and solve the network with $\sum_{i=1}^{N} B_i$ customers and calculate $\lambda(\sum_{i=1}^{N} B_i)$

S2: Estimate the parameters of the curve that passes through above calculated points.

S3: Calculate the unknown throughput points from the equations of these curves.

The critical step in the algorithm is finding the number of customers ,K^*, that produces the maximum throughput, $\lambda(K^*)$. Using the results of equivalencies between closed queueing networks with respect to the number of customers in the network (cf. Onvural and Perros [30]), K^* is determined approximately for cyclic networks and exactly for networks with exactly one node with infinite buffer capacity. Another drawback of the algorithm is solving the network numerically with K^* customers to calculate $\lambda(K^*)$. Still, the algorithm results in savings of 50 to 85 % of CPU time as compared to solving the network numerically for $K=1,\ldots,\sum_{i=1}^{N} B_i$ and produces accurate results.

Now, consider an exponential cyclic network, shown in Figure 1. The following lemma was proved in Onvural and Perros [31].

Lemma 8: In exponential cyclic networks, if the number of customers in the network ,K, is equal to the capacity of the network then the throughput of the network is equal to $1/E[\max(X_1,\ldots,X_N)]$ where X_i is the service time at node i. Furthermore, since X_i's are distributed exponentially, we have:

$$E[\max(X_1,\ldots,X_N)]=\int_0^{\infty}((1-\prod_{i=1}^{N}(1-e^{-\mu_i t}))\, dt.$$

For presentation purposes, let N=3, K=3 and $B_i=1$, i=1,2,3. Let X_i be the service time at node i and without loss of generality assume that $X_1 \le X_2 \le X_3$. Furthermore, assume that at t=0 all servers are busy working. Then at $t=X_3$, all three servers will become blocked and deadlock will occur. If we assume that deadlocks are detected immediately and resolved by instantaneously exchanging blocked customers then at $t=X_3$, customer at node 1 will go to node 2, customer at node 2 will go to node 3, and customer at node 3 will go to node 1. At this point in time, all servers will start a new service. The points in which all three servers start a new service are the renewal points and the throughput of the cyclic network is 1/(expected time between renewals) by definition.

IV.II. Mean Queue Lengths

The mean queue length of anode is an another primary performance measure of queueing networks. Let L_i be the mean queue length of node i. Then:

$$L_i = \sum_{j=1}^{\min B_i, K} jP_i(j) \qquad \text{and} \qquad \sum_{i=1}^{N} L_i = K$$

where, P_i(j) is the marginal probability of having j customers at node i.

It is discussed in Diehl [21] that the algorithm reported in Suri and Diehl [38,39] (which will be discussed in Section V) can be used for performance measures other than the throughput of the network. However, no numerical investigation for such performance measures has been done neither in Diehl [21] nor in Suri and Diehl [38,39]. Hence, to the best of our knowledge, the only algorithm to calculate the mean queue lengths of closed queueing networks with finite buffers under type 1 blocking is reported by Akyildiz [5]. His algorithm produces fairly accurate results. The approximation algorithm is based on the idea of analyzing the given blocking network with infinite buffer capacities and normalizing the non-feasible states which exceed the capacities of nodes. This is accomplished by the following procedure:

Let, $(k_1, \ldots, k_N)$ be the state of the network under consideration with infinite buffer capacities, where k_j is the number of customers at node j. Assuming the network has exponential servers, the product form solution of the non-blocking network is then used to calculate the marginal probabilities of nodes which are required to calculate the mean queue lengths. In case of general servers (cf. Akyildiz [6]), the algorithm is still applicable by first applying Marie's method [28]. When Marie's method converges, we have an exponential blocking network as above. Normalization of states is done as follows:

For any given state $(k_1, \ldots, k_V)$, if there exists a node i with $k_i > B_i$ then;

$$k_j = \begin{cases} B_j & \text{if } i = j \\ k_j + (k_i - B_i)\dfrac{e_j p_{ji}}{e_i(1-p_{ii})} & otherwise \end{cases}$$

Hence, if the capacity of a node is exceeded at some state for some node i, then the number of customers at that node is set to its capacity and the remaining customers are distributed to other nodes according to the routing probabilities (p_{ij}'s). e_i is the mean number of visits a customer makes to ith node and is given by eq. (7). Let the normalized state be given by $f(\underline{k})$ where the ith component of f, i.e. $f_i(\underline{k})$ is the number of jobs at station i after the normalization step. Then, $L_i = \sum_{\underline{k}\ feas} f_i(\underline{k})p(\underline{k})$, where $p(\underline{k})$ is the steady state joint queue length distribution of the network with infinite buffer capacities.

IV.III. Blocking network as an approximate BCMP node

Perros, Nilsson and Liu [34] developed a numerical procedure for analyzing exactly closed exponential queueing networks with finite buffers. The numerical procedure was then incorporated in an approximation algorithm for analyzing product form networks (i.e. with infinite buffer capacities) in which some of the buffers are finite. The approximation algorithm is based on Norton's theorem (cf. Chandy, Herzog and Woo [18]).

S0: Group all the finite queues and those infinite queues that are liable to getting blocked into a sub-network (blocking sub-network) and the remaining ones into an another subnetwork (non-blocking subnetwork).

S1: Analyze the non-blocking subnetwork (obtained from the original network by "shorting" the blocking subnetwork) as a product form network assuming n customers, where n=1,2,...,K. For each n obtain the steady state probabilities $p(\underline{l} \mid n)$, where $\underline{l} \in S_n$ is the state of the non-blocking sub-network and S_n is the set of all feasible states for a given n. Based on these results, calculate T(n), n=1,2,...,K.

S2: Construct a composite queue with a state dependent throughput equal to T(n), n=1,2,...,K. Now, in the original network substitute the non-blocking subnetwork by its composite queue. Analyze the reduced network numerically to obtain the marginal queue length probability distribution for each queue, assuming K customers in it.

S3: Let p_c(n), n=1,...,K be the marginal queue length probability that there are n units in the composite queue as calculated in S2. Then, from S1, we have $p(\underline{l})=p(\underline{l} \mid n)p_c(n)$, $n \in S_n$, and n=1,2,...,K. Using $p(\underline{l})$, marginal queue length distribution for each queue in the non-blocking subnetwork can be easily obtained.

Numerical investigation shows that the algorithm is very accurate for the throughput, mean queue lengths as well as the marginal queue length distributions of each node in the network.

The only drawback of the algorithm is solving the blocking network numerically which is time consuming for large networks.

V. Type 2 Blocking

While type 1 blocking was introduced in the context of production systems, type 2 blocking is used to model computer and telecommunications systems. The main difference between the two types of blocking mechanisms is that, in type 1, blocking occurs after service completion and in type 2, blocking occurs before service starts. In order to provide an intuitive explanation of the difference in definitions, we reinterpret what occurs during the service. In type 1 blocking, servers work on customers and the space is there to allow for smooth operation. In type 2 blocking, servers just move customers between spaces and do no other work on them. Hence, the lack of a space in the destination buffer forces the server to shut down (cf. Suri and Diehl [38]).

V.I. Type 2.1 Blocking Mechanism

In this case, a customer is not allowed to occupy the service space when the destination buffer is full. This blocking mechanism is introduced in open networks and, to the best of our knowledge, no study has been reported in the context of closed queueing networks. However, since type 2.1 blocking mechanism is equivalent to type 1 blocking mechanism in cyclic exponential networks, results reported for cyclic networks under type 1 blocking mechanism is readily applicable to cyclic networks under type 2.1 blocking mechanism after the buffer capacities are adjusted.

Furthermore, type 2.1 blocking mechanism may not be well defined for arbitrary topologies with arbitrary number of customers in it. For presentation purposes, consider the subnetwork shown in Figure 5.

Now, suppose that node k is full. Then nodes i and j are blocked and service spaces in front of servers i and j can not be occupied. When a departure from node k occurs, customers at nodes i and j will enter to the respective service spaces and service will start. While these customers are receiving service, let there be an arrival to node j so that node j becomes full. At this moment, if a departure from node i occurs causing node k to become full then we have an undefined case at node j. i.e., a destination node , k, of a full node, j, is full that forces the service space in front of server j to be used which is not permitted. Such problems can be handled by restricting the number of customers in the network, topology and/or buffer capacities.

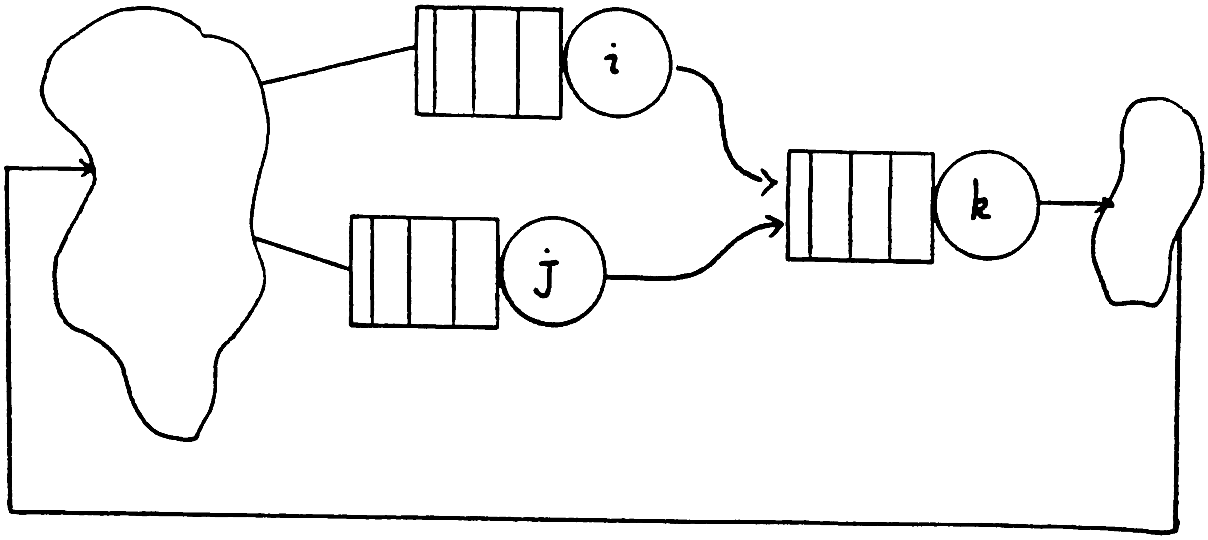


Figure 5: A sub − network

V.II. Type 2.2 Blocking Mechanism

Closed queueing networks under type 2.2 blocking were first studied by Gordon and Newell [23] in the context of cyclic exponential networks. Deadlock can not occur in this blocking mechanism because it can not be resolved. For presentation purposes, consider a cyclic network under type 2.2 blocking mechanism and let the number of customers in the network be equal to the capacity of the network. Then, each node is full, hence all servers are blocked. By definition, a server starts serving a customer if there is a space in the destination buffer, which is not possible if the network is full. Hence, the number of customers in the network should be less than its capacity, i.e. $K < \sum_{i=1}^{N} B_i$.

Now, we will discuss the concept of **holes** as introduced by Gordon and Newell [23]. Since the capacity of node j is B_j, let us imagine that this node consists of B_j cells. If there are i_j customers at node j, then i_j of these cells are occupied and $B_j - i_j$ cells are empty. We may say that these empty cells are occupied by holes. Then, the number of holes in the system is equal to $\sum_{i=1}^{N} B_i$-K. As the customers move sequentially through the network, the holes execute a counter sequential motion since each movement of a customer from the jth node to the (j+1)st node corresponds to the movement of a hole in the opposite direction. It is then shown that these two systems are duals. If a customer (a hole) at node j is blocked in one system then node j+1 has no holes (no customers) in its dual. To summarize the duality, let (B_i,μ_i) be the parameters of a node i and $((B_1,\mu_1),(B_2,\mu_2),...,(B_N,\mu_N))$ be a cyclic network. Then its dual is: $((B_1,\mu_N),(B_N,\mu_{N-1}),...,(B_2,\mu_1))$

Let p($\underline{n}$) and $p^D(\underline{n})$ are the steady state queue length distribution of a cyclic network and its dual with K and $\sum_{i=1}^{N} B_i - K$ customers respectively. $\underline{n}=(i_1,i_2, \ . \ . \ . \ , i_N)$ is the state of the network where i_j is the number of customers at node j, then:

$p(i_1, \ . \ . \ . \ , i_N) = p^D(B_1-i_1,B_N-i_N, \ . \ . \ . \ ,B_2-i_2)$ for all feasible states $\underline{n}$.

Note that, if the number of customers in the network is such that no node can be empty, then the dual network is a non-blocking network (i.e. the number of holes is less than or equal to the minimum buffer capacity) and has a product form solution. But then, from the arguments given above, the original network has a product form solution.

Approximation algorithms for cyclic networks under type 2.2 blocking was proposed by Suri and Diehl [38] and Onvural and Perros [31].

In particular, the algorithm given by Onvural and Perros [31] for type 1 blocking in Section IV.I. was also applied to approximate the throughput of cyclic networks under type 2.2 blocking. The following corollary is a consequence of duality in such networks.

Corollary 1: An exponential cyclic network under type 2.2 blocking mechanism with K customers has the same throughput as the network with $\sum_{i=1}^{N} B_i$-K customers.

Furthermore, Onvural and Perros [31] empirically observed that throughput of cyclic networks under type 2.2 blocking is non-decreasing as K increases from one to $\left\lfloor \sum_{i=1}^{N} B_i/2 \right\rfloor$. Hence, in their algorithm, it was assumed that the maximum throughput occurs at $K^* = \left\lfloor \sum_{i=1}^{N} B_i/2 \right\rfloor$. For $1 \le K \le \min(B_i)$, the network has a product form solution and the throughput can be calculated using one of the efficient algorithms for product form networks. After solving the network with $K^* = \left\lfloor \sum_{i=1}^{N} B_i/2 \right\rfloor$ customers and calculating the throughput, $\lambda(K^*)$, a curve was fitted that passes through known points, $\lambda(1),\ldots,$ $\lambda(min(B_i))$ and $\lambda(K^*)$ to estimate unknown throughput points.

Suri and Diehl [38] introduced the concept of "variable buffer size" and used it together with "flow-equivalent" approximations to approximate the performance measures of closed queueing networks with blocking.

Consider a cyclic network shown in figure 1 with at least one node with infinite buffer capacity. Furthermore, assume that service time at each node is distributed exponentially with rate μ_i, i=1,...,N. Without loss of generality, let node 1 has infinite buffer capacity.

In the flow-equivalent approach (cf. Chandy, Herzog and Woo [18], Chandy and Sauer [19]), all nodes other than node i, for some i, is replaced by a single server (i.e. composite server) with state dependent service rates, $\mu_i(l)$, where l is the number of customers in the composite queue. When this approach is used in networks with finite buffers, the buffer capacity of the composite queue plays an important role. If we use a buffer capacity of $B = \sum_{j=i+1}^{N} B_i$, this would overestimate the throughput, because node i can be blocked in the actual network with less than B customers in nodes i+1 to N. If we use $B = B_{i+1}$, this will underestimate the throughput because when there are B_{i+1} customers at node i+1 to B, not all of them need to be at node i+1. Thus, server i, S_i, "sees" a finite buffer of size k in the composite queue, for some fraction of time. The variable buffer size model introduced by Suri and Diehl [38] is an attempt to capture this view seen by S_i.

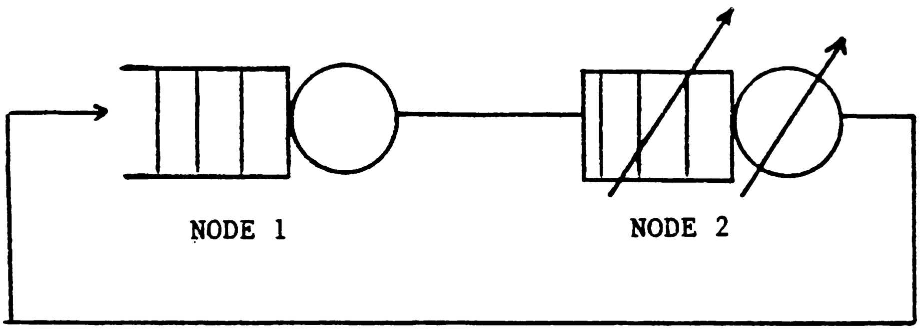


Figure 6:A variable buffer-size model

Let, p(k | K) be the fraction of time composite queue behaves like a k-buffer node. Given, the fixed buffer-size and state dependent service rates, two-node network has a product form solution hence it can be solved efficiently. If p(k | K)'s are known then the performance measures of the original network can be calculated as a weighted sum of the performance measures of the two-node network. The complete algorithm to calculate the throughput of cyclic networks is given below.

ALGORITHM TO SOLVE CYCLIC NETWORK
BEGIN
/* INPUT $(K, N, \mu_i, B_i, i=1,\ldots,N;$ */
/* INITIALIZE FIRST VIEW */
  FOR J=1 TO K DO
   BEGIN
    $\lambda_N(J)=\mu_N$
    FOR k=1 TO J DO
     IF $k=B_N$ then $p_N(k \mid J)=1$ ELSE $p_N(k \mid J)=0$;
   END;
/* MAIN LOOP:COMPUTE VIEW S_{N-1} THRU 1 */
  FOR I=N-1 DOWNTO 1 DO
   BEGIN
    FOR J=1 to K DO
     BEGIN
      Solve network VB_i obtaining $X^{VB_i}(J)$ and $P_{1,l}^{VB_i}(J)$; (Figure 6)
      $\lambda_i(J)=X^{VB_i}(J)$;
      FOR k=1 TO J-1 DO $p_i(k \mid J)=P_{1,k-B_i}^{VB_i}(J)$;
      $p_i(J \mid J)=\sum_{j=0}^{B_i} P_{1,J-B_i+j}^{VB_i}(J)$;
    END;
   END;
/* OUTPUT */
  FOR J=1 TO K DO OUTPUT($\lambda_1(J)$);
END.

The approximation algorithm is based on the idea that the ith server can view all the downstream nodes i+1 to N in terms of a single finite queue which is the flow equivalent queue of the downstream servers. In particular, node 2 of the variable buffer size model (figure 6) is this flow equivalent queue. The service rate at node 1 of the variable buffer size model is that of queue i, i.e. μ_i. The total number of customers is varied from 1 to K for each node i. The weights $p_i(k \mid K)$ are then obtained by calculating the probabilities that there are b_i customers in the first node and k-b_i in the second node.

The algorithm proceeds in this fashion moving upstream until all queues have been considered. The algorithm is easily started seeing that the (N-1)st queue sees a flow equivalent queue of the downstream queues which is nothing else but the Nth queue itself since the destination node of the Nth node (node 1) has infinite buffer capacity. The algorithm is accurate and fast. Validation tests are restricted to three-node cyclic networks with at least one node with infinite buffer capacity and comparisons were made only for the throughput. A similar algorithm is given for cyclic networks under type 1 blocking. It is discussed in Diehl [21] that the algorithm can be used to approximate other performance measures and it is also applicable to arbitrary topologies as well as open-networks. No validation tests have been reported for such cases.

VI. Type 3 Blocking

This type of blocking was introduced by Caseau and Pujolle [17] in open tandem networks and by Pittel [36] in reversible closed queueing networks. It was initially used to model communication networks where a packet is retransmitted due to the fact that the destination buffer was full. Recently, it is used to model flexible manufacturing systems (cf Yao and Buzacott [40,41,42,43]).

VI.I. Type 3.1 Blocking Mechanism

We first note that type 3.1 blocking mechanism is equivalent to type 2.2 blocking mechanism in of cyclic exponential networks. Thus, exact and approximate results discussed for type 2.2 blocking section V.II are readily applicable to this blocking mechanism. For general topologies, if all nodes that are subject to blocking in a closed exponential network have exactly one destination node. then type 3.1 blocking is equivalent to type 3.2 blocking. Hence, results that are discussed next in Section VI.II are applicable to this blocking mechanism. Other than these equivalencies, there are no results reported for this blocking mechanism in closed queueing networks.

VI.II. Type 3.2 Blocking Mechanism

This section of the survey is taken from Perros [34]. Let us consider a closed queueing network with parameters as given in table 1 under type 3.2 blocking mechanism. A customer at the ith node requires an exponentially distributed service with mean $1/\mu_i$. Let $f_i(n_i)$ be the rate at which the server at the ith node works when there are n_i customers in the node, i=1,...,N. We have $f_i(n_i) > 0$ if $n_i > 0$ and $f_i(n_i)=0$ if $n_i=0$. Furthermore, let $b_j(n_j)$ be the probability that a customer will be admitted to node j when there are n_j customers in the node. Note that in type 3.2 blocking, $b_j(B_j)=0$ and $b_j(k)=1$ for $0 \le k < B_j$, j=1,...,N. Let $\pi(i_1, \ldots, i_N)$ be the steady state queue length distribution of a closed queueing network where i_j is the number of customers at node j, $\sum_{i=1}^{N} i_j = K$, and $i_j \le B_j$, j=1,...,N. Now, if the routing matrix (p_{ij}'s) is reversible then there exists positive λ_i, i=1,...,N which satisfy the following equation (cf. Kelly [26]):

$$\lambda_i p_{ij} = \lambda_j p_{ji}$$

for all i and j. In this case, it is shown that (cf. Pittel [36], Hordijk and Van Dijk [25]):

$$\pi(i_1, \ldots, i_N) = C \prod_{j=1}^{N} \prod_{k=1}^{i_j} \frac{\lambda_j}{\mu_j(k) f_j(k)} \, b_j(\text{k-1})$$

where C is a normalizing constant.

Therefore, if the routing matrix is reversible, the queueing network described above has a product form steady state solution. When the routing matrix is not reversible, the above queueing network can still have a product form solution in the following two cases (cf. Hordijk and Van Dijk [25]):

a) The probability $b_j(n_j)$ that a customer will be admitted to node j is constant, independent of n_j. That is $b_j(n_j)=b_j$, j=1,...,N. Note that this definition of blocking does not correspond to any of the blocking mechanisms defined in Section II.

b) The rate $f_i(n_i)$ at which server i works is constant, independent of n_i. That is $f_i(n_i)=f_i$. Note that, in cyclic networks, this result is immediate from the equivalency of type 2.2 and type 3.2 blocking mechanisms since a constant rate, f_i, would mean no node in the network can be empty. Akyildiz [4] extended this result to include at most one empty station.

We, now, consider the case where the routing probabilities from node i to node j is state dependent (cf. Yao and Buzacott [40]). In particular let us consider the queueing network studied above in this section assuming that p_{jk} depends on i_j and i_k as follows:

$$p_{jk}(i_j,i_k)=\Phi_j(i_j)\Psi_k(i_k)$$

where $\Phi_j(.)$ and $\Psi_k(.)$ are arbitrary functions such that $\Phi_j(i_j)>0$ if $i_j>0, \Psi_k(i_k)>0$ if $i_k>0$ and $\Phi_j(0)=0$ and $\Psi_k(B_k)=0$. Under these state dependent routing probabilities, the queueing network is reversible and has the following product form steady state solution.

$$\pi(i_1,\ldots,i_N)=C\prod_{j=1}^{N}\prod_{k=1}^{i_j}\frac{\Psi_j(k-1)}{\mu_j(k)f_j(k)\Phi_j(k)}$$

where C is a normalizing constant.

The effect of this state dependent routing is as follows. A customer upon completion of its service at node i will probabilistically join any of the destination nodes which, at that moment, are not full. If all of the destination nodes are full then the service will be repeated at node i (type 3.2 blocking).

We note that the routing probability $p_{jk}(i_j,i_k)$ should satisfy $\sum_k p_{jk}(i_j,i_k)=1$ for all j. From this, we have:

$\sum\Phi_j(i_j)\Psi_l(i_l)=1$, equivalently we have:

$$\Phi_j(i_j)=1/\sum_l \Psi_l(i_l)=1$$

Now. let $\Psi_j(i_j)=B_l-i_l$, then:

$$\Phi_j(i_j)=1/\sum_{\substack{l \\ l\neq j}} B_l\text{-}(K\text{-}(i_j))$$

Thus, we have the following "probabilistic shortest queue" routing (Yao and Buzacott[40]):

$$p_{jk}(i_j,i_k)=\frac{B_k-i_j}{\sum\limits_{\substack{l=1 \\ l\neq j}} B_l-(K-i_j)}$$

for $j \neq k$. In this type of routing, a job may join a node which has the shortest queue with the highest probability. A job never joins a node which is full hence no blocking can take place in the network. This is an extension of type 3.2 blocking to non-blocking networks with reversible routing.

Yao and Buzacott [43] reported an approximation algorithm for analyzing closed queueing networks under type 3.2 blocking. In addition to the parameters as given in table 1, assume that each queue is served by c_i servers. Service times are assumed to follow arbitrary coxian distributions. The topology of the network is such that if each service distribution is substituted by an exponential distribution with the same mean as the coxian server, then the resulting exponential network is reversible and has a product form solution. The approximation algorithm is based on the notion of exponentialization. The main steps of the algorithm as follows:

S0: For each node i, substitute its service distribution with the same rate $\mu_i(n_i)$ as the coxian server, $n_i=0,\ldots,B_i$; $i=1,\ldots,N$

S1: Solve the resulting reversible network to obtain the marginal queue length distribution $p_i(n_i)$ for each node i.

S2: Derive state dependent arrival rate $\lambda_i(n_i)$ to each node i, i=1,...,N using:

$\lambda_i(n_i) = \mu_i(n_i + 1)p_i(n_i - 1)/p_i(n_i)$, $n_i = 0,\ldots,B_i$

S3: Analyze each node in isolation as a $\lambda_i(n_i)/G/c_i/B_i$ queue, where $\lambda_i(n_i)$ are obtained from step 2. Remaining parameters c_i, G, B_i are the same as in the original network. For each node, obtain marginal queue-length distribution, $q_i(n_i)$, n_i=0,...,B_i and i=1,...,N.

S4: Derive state dependent service rates, $v_i(n_i)$'s as follows:

$v_i(n_i) = \lambda_i(n_i - 1)q_i(n_i)/p_i(n_i)$, $n_i = 0,\ldots,B_i$

n_i=0,...,B_i and i=1,...,N.

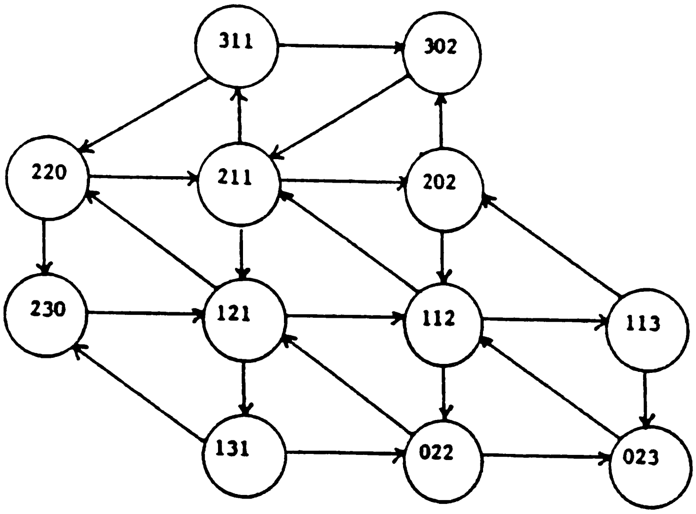
S5: If $\max_{i,n_i} |v_i(n_i) - \mu_i(n_i)| < \epsilon$ then STOP else set $\mu_i(n_i) = v_i(n_i)$ for all n_i=0,...,B_i and i=1,...,N and go back to S1.

The algorithm can be used to obtain marginal queue length probability distribution. Validation examples showed that the accuracy of the algorithm is satisfactory.

VII. Special Case

In this section, we will discuss the concept of indistinguishable nodes as introduced by Onvural and Perros [30] and Persone and Grillo [20] in symmetric cyclic networks.

Consider exponential cyclic networks , shown in Figure 1, with parameters B_i=B $< \infty$ and $\mu_i = \mu$, i=1,...,N. Then, $1 \le K \le NB$. The algorithm given in this section utilizes an aggregate state space obtained from the original state space of the network after it is reduced by a factor of N. For presentation purposes, consider a cyclic network under type 1 blocking with B=2, K=4 and N=3. The state space has the following structure with all transition rates equal to μ.



Solving the system numerically (cf. Perros, Nilsson and Liu [11]), we have:

$$P(2,2,0)=P(0,2,2)=P(2,0,2)=0.071429$$

$$P(2.3,0)=P(0,2,3)=P(3,0,2)=0.11905$$

$$P(2,1,1)=P(1,2,1)=P(1,1,2)=0.095238$$

$$P(3,1,1)=P(1,3,1)=P(1,1,3)=0.047619$$

This result is not surprising seeing that nodes are indistinguishable. In view of this, let us define the following classes, where a state is a member of a class if that state has the same steady state probability as all the other states in the same class.

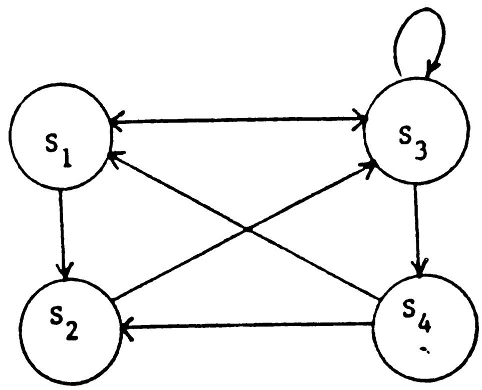
$$S_1=\{(2,2,0),(0,2,2),(2,0,2)\}$$

$$S_2=\{(2,3,0),(0,2,3),(3.0.2)\}$$

$$S_3=\{(2,1,1),(1,2,1),(1,1,2)\}$$

$$S_4=\{(3,1,1),(1,3,1),(1,1,3)\}$$

Note that $i_j=3$ $(=B_j+1)$ denotes that node j-1 is blocked by node j. Then, we have the following state space structure for these equivalence classes with all transition rates equal to μ.



Solving this system numerically, we have:

$P(S_1)=0.214287$; $P(S_2)=0.35715$; $P(S_3)=0.28571$; $P(S_4)=0.142857$

Furthermore, $P(S_i)= \sum_{(i_1,i_2,i_3)\in S_i} P(i_1,i_2,i_3)$. i=1,...4. Hence, to solve the original network, we can form the equivalence classes ,S_i, create the rate matrix for these classes and solve the system. Then we can obtain the queue length distribution of the original network. The following algorithm summarizes this procedure.

ALGORITHM

1.Generate the equivalence classes, S_i, and set up the rate matrix.

2.Solve the system to obtain $P(S_i)$.

3.Calculate the normalizing constant, G_K, for the original network as follows:

$$G_K = \sum_{i=1}^{S} R_i P(S_i)$$

where S is the number of equivalence classes and R_i is the number of states in equivalence class i.

4.$P(i_1,..,i_N)=G_K^{-1}P(S_i)$ where $(i_1,...,i_N) \in S_i$

Finally, we note that, although the concept of indistinguishable nodes are discussed in cyclic networks under type 1 blocking, it is applicable to other blocking mechanisms in exponential cyclic networks.

Acknowledgement

I am indebted to Dr. Harry G. Perros for introducing the concept of blocking to me. During my studies, not only he did motivate and encourage my work but also created the environment that made it possible.

REFERENCES

[1] **Akyildiz, I.F.**, "Exact Product Form Solution for Queueing Networks with Blocking", IEEE Transactions on Computers, 1, 121-127, Jan. 1987

[2] **Akyildiz, I.F.**, "On the Exact and Approximate Throughput Analysis Of Closed Queueing Networks with Blocking", to appear in IEEE Transactions on Software Engineering

[3] **Akyildiz, I.F. and Kundu, S.**, "Buffer Allocation in Deadlock Free Blocking", CS Dept, (1986), Lousiana State University

[4] **Akyildiz, I.F. and Von Brand, H.**, "Duality in Open and Closed Markovian Queueing Networks with Rejection Blocking", CS Dept., 87-011, (1987), Lousiana State University

[5] **Akyildiz, I.F.**, "Mean Value Analysis for Closed Queueing Networks with Blocking", to appear in IEEE Transactions on Software Engineering

[6] **Akyildiz, I.F. and Sieber, A.**, "General Closed Queueing Networks with Blocking", CS Dept., 87-015, (1987), Lousiana State University

[7] **Altiok, T., and Stidham, Jr., S.**, "A Note on Transfer Lines with Unreliable Machines, Random Processing Times and Finite Buffers", IIE Trans., 14, 125-127, (1982)

[8] **Altiok, T. and Perros, H.G.**, "Open Networks of Queues with Blocking: Split and Merge Configurations", CS Rep., 83-10, NC State Univ., (1983)

[9] **Altiok, T. and Perros, H.G.**, "Approximate Analysis of Arbitrary Configurations of Queueing Networks with Blocking", AIIE Transactions, March 1986

[10] **Bacharov, P.P. and Albores, F.K.**, "On Two Stage Exponential Queueing Systems with Internal Losses or Blocking", Problems of Control and Information Theory, 9, 365-379, (1980)

[11] **Balsamo, S. , Persone V. De Nitto, and Iazeolla, G.**, "Some Equivalencies of Blocking Mechanisms in Queueing Networks with Finite Capacity", Manuscript, Dipartimento di Informatica, Universita di Pisa, Pisa Italy, (1986)

[12] **Balsamo, S. and Iazeolla, G.**, "Some Equivalence Properties for Queueing Networks with and without Blocking", Performance'83, Agrawala and Tripathi (Eds.), North Holland, 351-360, (1983)

[13] **Baskett, F., Chandy, K.M., Muntz, R.R., and Palacios, F.G.**, "Open, Closed and Mixed Networks of Queues with Different Classes of Customers", J. of ACM, 22-2, 249-260, (1975)

[14] **Boxma, O. and Konheim, A.**, "Approximate Analysis of Exponential Queueing Systems with Blocking", Acta Informatica, 15, 19-66, (1981)

[15] **Bruell, S.C. and Balbo, G.**, "Computational Algorithms for Closed Queueing Networks", Operating and Programming Systems Series, Peter J. Denning, Editor

[16] **Buzen, J.P.**, "Computational Algorithms for Closed Queueing Networks with Exponential Service Times", Comm. ACM, 16-9, 527-531, (1973)

[17] **Caseau, P. and Pujolle, G.**. "Throughput Capacity of a Sequence of Transfer Lines with Blocking Due to Finite Waiting Room", IEEE Trans. Software Eng., 5, 631-642, (1979)

[18] **Chandy, K.M., Herzog, U. and Woo, L.**, "Parametric Analysis of Queueing Networks", IBM J. Res. Dev., 19, 43-49, Jan. 1975

[19] **Chandy, K.M., and Sauer, C.H.**, "Approximate Methods for Analyzing Queueing Network Models of Computing Systems", Comp. Surveys, 10-3, 281-317, (1978)

[20] **De Nitto Persone, V. and Grillo, D.**, "Managing Blocking in Finite Capacity Symmetrical Ring Networks", 3rd Conf. on Data and Communication Systems and Their Performance, 1987

[21] **Diehl, G.W.**, "A Buffer Equivalency Decomposition Approach to Finite Buffer Queueing Networks", Ph.D. Thesis, Eng. Sci., Harvard University, (1984)

[22] **Gershwin, S. and Berman, U.**, "Analysis of Transfer Lines Consisting of Two Unreliable Machines with Random Processing Times and Finite Storage Buffers", AIIE Trans., 13, No:1, 2-11, (1981)

[23] **Gordon, W.J. and Newell, G.F.**, "Cyclic Queueing Systems with Restricted Length Queues", Oper. Res., 15, 266-278, (1967)

[24] **Gordon, W.J. and Newell, G.F.**, "Closed Queueing Systems with Exponential Servers", Oper. Res., 15, 254-265, (1967)

[25] **Hordijk, A. and Van Dijk, N.**, "Networks of Queues with Blocking", Performance'81, Klystra (Ed.), 51-65, North Holland, (1981)

[26] **Kelly, F.P.**, "Reversibility and Stochastic Networks", John Wiley and Sons, (1979)

[27] **Latouche, G. and Neuts, M.F.**, "Efficient Algorithmic Solutions to Exponential Tandem Queues with Blocking", SIAM J. Alg. Disc. Math., 1, 93-106, (1980)

[28] **Marie, R.**. "An Approximate Analytical Method for General Queueing Networks", IEEE Tran. Soft. Eng., 5, No:5, 530-538, (1979)

[29] **Onvural, R.O. and Perros, H.G.**, "On Equivalencies of Blocking Mechanisms in Queueing Networks with Blocking", OR Letters, 5, No:6, 293-298, (1986)

[30] **Onvural, R.O. and Perros, H.G.**, "Some Exact Results on Closed Queueing Exponential Queueing Networks with Blocking", CS Dept, Tech. Report, NCSU, (1986)

[31] **Onvural, R.O. and Perros, H.G.**, "Throughput Analysis in Closed Queueing Networks with Finite Buffers", CS Dept., 87-03, NCSU, (1987)

[32] **Onvural, R.O.**, "Closed Queueing Networks with Finite Buffers", Ph.D. Thesis, CSE/OR. NCSU. (1987)

[33] **Perros, H.G.**, "A Survey of Queueing Networks with Blocking (Part I)", Dept. of CS, 86-04, NC State Univ., (1986)

[34] **Perros, H.G., Nilsson, A.A. and Liu, Y.C.**, "Approximate Analysis of Product Form Type Queueing Networks with Blocking and Deadlock", CCSP, NC State Univ., (1986)

[35] **Perros, H.G.**, "A Survey of Queueing Networks with Blocking (Part II)", In preparation

[36] **Pittel, B.**, "Closed Exponential Networks of Queues with Saturation: The Jackson Type Stationary Distribution and Its Asymptotic Analysis", Math. Oper. Res., 4, 367-378, (1979)

[37] **Shantikumar, G.**, Personal Conversation

[38] **Suri, R. and Diehl, G.W.**, "A Variable Buffer Size Model and its Use in Analytical Closed Queueing Networks with Blocking", Proc. ACM SIGMETRICS on Measurement and Modelling of Computer Systems, 134-142, (1984)

[39] **Suri, R. and Diehl, G.W.**, "A Variable Buffer Size Model and its Use in Analytical Closed Queueing Networks with Blocking", Management Science, 32-2, 206-225, (1986)

[40] **Yao, D. D. and Buzacott, J.A.** "Modelling a Class of State Dependent Routing in Flexible Manufacturing Systems", Annals of Oper. Res., 3, 153-167, (1985)

[41] **Yao, D. D. and Buzacott, J.A.** "Queueing Models for Flexible Machining Station Part I:Diffusion Approximation", Eur. J. of Oper. Res., 19, 233-240, (1985)

[42] **Yao, D. D. and Buzacott, J.A.** "Queueing Models for Flexible Machining Station Part II: The Method of Coxian Phases", Eur. J. of Oper. Res., 19, 241-252, (1985)

[43] **Yao, D. D. and Buzacott, J.A.** "The Exponentialization Approach to Flexible Manufacturing System Models with General Processing Times", Eur. J. of Oper. Res., 24, 410-416, (1986)