# Survey of Closed Queueing Networks with Blocking

RAIF O. ONVURAL*

*BNR, Department 3N23, P.O. Box 13478, Research Triangle Park, North Carolina 27709-3478*

Closed queueing networks are frequently used to model complex service systems such as production systems, communication systems, computer systems, and flexible manufacturing systems. When limitations are imposed on the queue sizes (i.e., finite queues), a phenomenon called *blocking* occurs. Queueing networks with blocking are, in general, difficult to treat. Exact closed form solutions have been reported only in a few special cases. Hence, most of the techniques that are used to analyze such queueing networks are in the form of approximations, numerical analysis, and simulation. In this paper, we give a systematic presentation of the literature related to closed queueing networks with finite queues. The results are significant for both researchers and practitioners.

Categories and Subject Descriptors: C.4 [**Computer Systems Organization**]: Performance of Systems—*modeling techniques, performance attributes*; I.6.3 [**Simulation and Modeling**]: Applications; D.2.8 [**Software Engineering**]: Metrics—*performance measures*; D.4.8 [**Operating Systems**]: Performance—*queueing theory, simulation, stochastic analysis*; G.m [**Mathematics of Computing**]: Miscellaneous—*queueing theory*

General Terms: Performance, Modeling

Additional Key Words and Phrases: Blocking, finite buffer capacities, Markov model, queueing networks

## INTRODUCTION

System performance has been 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 such systems is judged by the degree to which performance objectives are met. Thus, tools and techniques for predicting performance measures are of great interest.

Queueing theory was developed to understand and predict the behavior of real life systems. Conceptually, the simplest queueing model is the single queueing system illustrated in Figure 1. The system models the flow of customers as they arrive, wait in the queue if the server is busy serving another customer, receive service, and eventually leave the system.

To describe the behavior of a queueing system in time, five basic characteristics of the process need to be specified: (1) the arrival pattern, (2) the number of servers, (3) the service pattern, (4) the service discipline, and (5) the system capacity. The arrival pattern, or *input*, to a queueing system is often measured in terms of the average number of arrivals per some unit of time, called the mean arrival rate. If the arrival pattern is deterministic, the arrival

---

* This work was done while the author was at the Department of Computer Science and the Center for Communications and Signal Processing, North Carolina State University, Raleigh, North Carolina 27695-8206.

CONTENTS



process is fully determined by the mean arrival rate. If the arrival pattern is random, further characterization is required in the form of the probability distribution associated with the process. The number of service channels refers to the *number of parallel servers* that can service customers simultaneously. The rate at which customers are served is called the mean *service rate*. In the case of deterministic service processes, specification of the mean service rate is sufficient to describe the process, whereas if the process is random its probability distribution needs to be specified. The manner by which customers are selected for service when a queue has formed is referred to as the *service discipline*. Finally, the *system capacity* is the upper limit on the number of customers (waiting for and receiving service) in the system. Kendall [1953] introduced the notation $A/B/X/Y/Z$ to describe the queueing process of a single queueing system, where $A$ indicates the arrival pattern, $B$ the service pattern, $X$ the number of parallel servers, $Y$ the system capacity, and $Z$ the service discipline. For example, D/D/1/$\infty$/FCFS describes a single queueing system with deterministic arrival and service processes, one server, infinite system capacity (i.e., there is always a space in the queue for arriving customers), and first come first served (FCFS) service discipline.

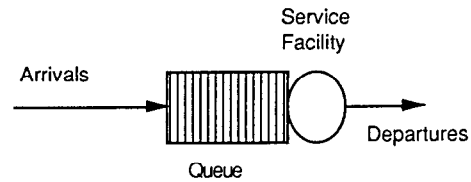


**Figure 1.** Single queueing system.

The most common queueing models assume that interarrival and service times obey the exponential distribution, or, equivalently, the arrival and the service rates follow a Poisson distribution. Consider an arrival process $\{N(t), t \geq 0\}$, where $N(t)$ denotes the total number of arrivals up to time $t$, with $N(0) = 0$, which satisfies the following assumptions [Gross and Harris 1974]:

(i) The probability that an arrival occurs between time $t$ and $\Delta t$ is equal to $\lambda \Delta t + o(\Delta t)$, where $\lambda$ is a constant, $\Delta t$ is an incremental element, and $o(\Delta t)$ denotes a quantity that becomes negligible as $\Delta t$ goes to zero.

(ii) Probability that there is more than one arrival between $t$ and $t + \Delta t$ is $o(\Delta t)$.

(iii) The numbers of arrivals in nonoverlapping intervals are statistically independent.

Let $p_n(t)$ be the probability of $n$ arrivals in a time interval $t$. Under the three assumptions above, we have $p_n(t) = (\lambda t)^n e^{-\lambda t}/n!$. This distribution is referred to as the Poisson distribution with rate $\lambda$. If $N(t)$ is Poisson with rate $\lambda$, then the time between arrivals is exponentially distributed with mean $1/\lambda$; that is, let $T$ be the random variable "time between arrivals," then: $\Pr\{T \leq t\} = 1 - e^{-\lambda t}$. One of the interesting properties of the exponential distribution is the Markovian (also called memorylessness) property, which states that the probability that a customer currently in service is completed at some future time $t$ is independent of how long the customer has already been in service.

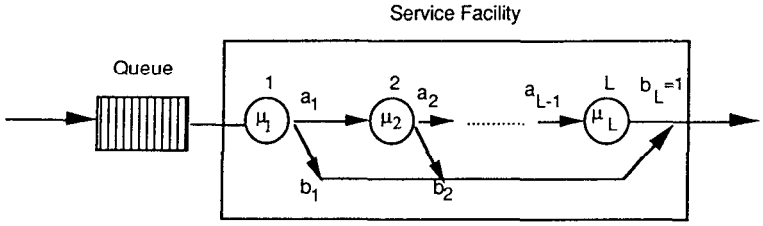


**Figure 2.** Queueing system with a Coxian server with $L$ stages.

The *variance* of service time [var(ST)] gives a rough measure of spread. In particular, if var(ST) is small, then the service times of customers are close to the mean with a little variation. For example, if the service times are deterministic, then var(ST) = 0 (i.e., no variation). If the service times are exponentially distributed, then their variance is equal to the mean service time. Conversely, a large variance indicates that the service times of customers are widely spread; hence there is a large variation from the mean. Finally, the *squared coefficient of variation* of the service time, $c_i^2$, gives a rough measure of the spread normalized over the square of the mean service time; that is, $c_i^2$ = var(ST)/(mean(ST))$^2$ [Solomon 1983]. We now define a family of probability distributions that are general and will allow us to relax the assumption that the service and the interarrival times are exponentially distributed. The idea is to use a simple exponential network to represent the service time required from a single server [Cox 1955].

Consider the service facility of the single queueing system illustrated in Figure 2. There can be at most one customer in nodes 1 to $L$ at any time. Customers enter the service via node 1. The service time at node $m$ is exponentially distributed with mean $1/\mu_m$. A customer completing its service at node $m$ leaves the system with probability $b_m$ or proceeds to node $m + 1$ with probability $a_m$ ($b_m + a_m = 1$, $m = 1, \ldots, L - 1$). After node $L$, the customer leaves the system with probability 1. This service distribution is referred to as a Coxian distribution with $L$ stages. Any probability distribution function can be approximated arbitrarily closely by Coxian distribution functions. Hence, an arbitrary distribution that does not have the Markovian property can be approximated by a Coxian distribution that has the Markovian property [Cox 1955].

A queueing process with Poisson arrivals, exponentially distributed service times, one server, with capacity B, and first come first served service discipline is referred to as an M/M/1/B/FCFS queue, where M stands for Markovian (memoryless). Let $\pi_n(t)$, $n \in S$, be the probability that there are $n$ customers at time $t$ in an M/M/1/B/FCFS queue, where $n$ and $S$ are, respectively, referred to as the *state* and the *state space* of the system. For the above queueing system, we have $S = \{0, 1, \ldots, B\}$. Furthermore, let us assume that the system state ($n$) eventually becomes independent of the initial state so that no matter what time we query the system, the probability of finding $n$ customers in the system remains constant, independent of time $t$. Then, $\pi_n$ is referred to as the *steady-state* probability of having $n$ customers in the system.

The *transition rate diagram* of a process is a graphical representation of the transitions between the states of the process. The directed arcs between the states denote the one-step transitions of going from one state to another. For example, transitions out of state $n$ of an M/M/1/B/FCFS queue occurs either to state $n + 1$ with an arrival (with rate $\lambda$) or to state $n - 1$ with a departure (with rate $\mu$). Its transition rate diagram is given in Figure 3.

The *global balance equations* equate the total rate out of state $n$ to the total rate into state $n$, for each state $n \in S$. The global balance equations of the

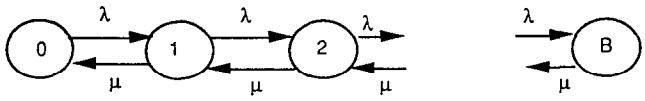


**Figure 3.** Transition rate diagram of an M/M/1/B/FCFS queue.

M/M/1/B/FCFS queue can be easily written from the transition rate diagram given in Figure 3:

$$\lambda\pi_0 = \mu\pi_1$$

$$(\lambda + \mu)\pi_n = \lambda\pi_{n-1} + \mu\pi_{n+1}, \qquad n = 1, \ldots, B-1$$

$$\mu\pi_N = \lambda\pi_{N-1}$$

In this system, there are $B + 1$ unknowns ($\pi_n$'s) and $B + 1$ equations of which only $B$ of them are linearly independent. As $\pi_n$ is a probability distribution, the sum of all $\pi_n$'s over its state space should add up to 1; that is, $\sum_{n=0}^{B} \pi_n = 1$. This equation is referred to as the *normalization equation.* Using any $B$ of the above $B + 1$ equations together with the normalization equation, there are $B + 1$ linearly independent equations, which can be solved to obtain $\pi_n$'s. Writing these equations in matrix form, we have

$$Q\underline{\pi} = b,$$

where

$$b = (0, \ldots, 0, 1)$$

is a $B + 1$ vector with 0's in positions 1 to $B$ and 1 at position $B + 1$;

$$\underline{\pi} = (\pi_0, \ldots, \pi_B)$$

is a $B + 1$ vector of steady-state queue length probabilities; and

$$Q = \begin{bmatrix} -\lambda & \mu & & & & \\ \lambda & -(\lambda+\mu) & \mu & & & \\ & \lambda & -(\lambda+\mu) & \mu & & \\ & & \ddots & \ddots & \ddots & \\ 1 & 1 & \cdots & 1 & 1 & \end{bmatrix}$$

is a $(B + 1) \times (B + 1)$ matrix referred to as the *rate matrix* of the process.

We note that in this rate matrix, we use the first $B$ of the above $B + 1$ equations and replaced the $B + 1$st equation with the normalization equation.

Customers in real life systems usually require several different services provided by different servers and may have to wait in several different queues before receiving the required services. Such complex service systems can be modeled by defining a network of single queueing systems, referred to as a *queueing network.* A queueing network can be thought of as a connected directed graph whose nodes represent the service centers. The arcs between those nodes indicate the one-step moves that customers may make from service center to service center. The set of nodes and the set of arcs that connect the nodes are referred to as the *topology* of the network. The route that a customer takes through the network may be deterministic or random. Customers may be of different types and may follow different routes through the network. Each node has a queue associated with it. To complete the definition of a queueing network, the assumptions must be specified for the parameters of each node (i.e., the number of servers, the service discipline, the node capacity, and the service time distribution).

Queueing networks can be classified as open or closed. In an open model, customers enter the network from outside, receive service at one or more nodes, and eventually leave the network. Figure 4 illustrates an open network.

To model a system using an open network assumes that the arrivals to the system occur from an infinite population of customers. That is, the rate at which customers arrive to the network is independent of the number of customers already in the system. If the user population is finite, then those already in the system are no longer candidates for entering the network. Hence, as the number of customers in the network increases, the available population

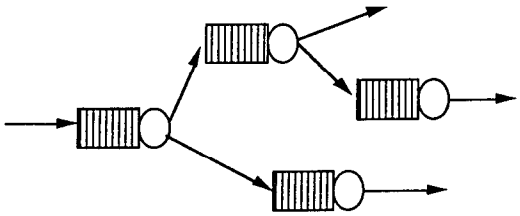

**Figure 4.** Open network.

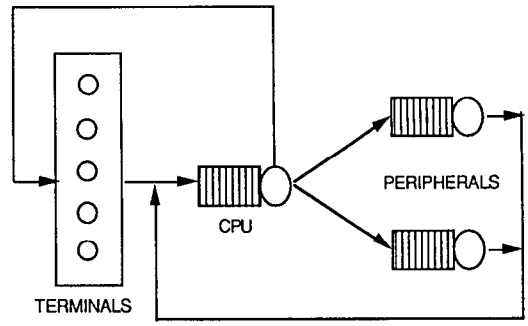


**Figure 5.** Model of a time-sharing system.

dwindles and the arrival rate falls off because of the reduced population that can generate arrivals to the network. Most real life systems have finite input populations. For example, in time-sharing systems, the number of jobs is limited by the number of terminals, so that the total number of jobs is bounded. As another example, in multiprogramming systems, the degree of multiprogramming is limited by the memory size. Similarly, in communication networks, the number of unacknowledged packets in a region of the network is limited by the window size. Open networks may not be used to model such systems in which the nature of the arrival process depends strongly on the number of customers already in the system [Kleinrock 1976].

In closed queueing networks, there is a fixed population of customers circulating in the network; that is, no arrivals to or departures from the network are allowed. Modeling real life systems as closed queueing networks is based on the assumption that the number of customers in the system is bounded. As an example, consider the simplistic view of a time-sharing system illustrated in Figure 5.

The system consists of a central processing unit (CPU) and two peripherals. Each job in the system is associated with one of the terminals, hence the number of jobs in the system is equal to the number of terminals. A job generated by a terminal goes to the CPU. Upon receiving its service, the job is either completed and goes back to the terminals node or requests an input/output operation and joins one of the peripheral devices node with respective probabilities. Upon completion of its service at the peripheral device, the job always goes back to the CPU node.

Closed queueing networks are also used to model service systems where the number of customers in the system is constant for a long period of time and there is always a customer waiting to enter whenever a departure occurs from the system. As an example, consider the simplified view of a packet switched network with fixed routing as illustrated in Figure 6. A physical network path is set up for each user session and is released when the session is terminated. End-to-end flow control is exercised to prevent buffer congestion at the exit node due to the fact that remote sources are sending traffic at a higher rate than can be accepted by the hosts fed by the exit node [Reiser 1979]. Sliding window strategies are among the most popular forms of end-to-end flow control. In this scheme, the number of packets that can be outstanding without an acknowledgment between any source–destination pair is constrained to be no more than some positive integer $w$, called the window size.

If the network is operating in a high load condition, then the source can be assumed to have a backlog of packets ready to send into the network as long as the window size allows. In this case, when a packet is delivered to the destination, a new packet immediately enters the network. Hence, the system can be modeled as a closed network with $w$ customers in it.

Queueing networks have been studied in the literature under a multiplicity of assumptions. After the pioneering work of Jackson [1963], the development and analysis of one of the most general queueing

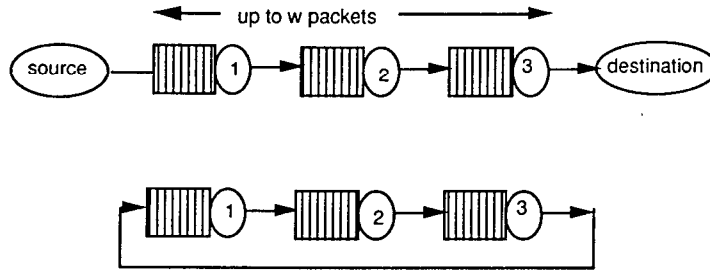


**Figure 6.** Network path with a window flow control and equivalent closed queueing model.

networks was due to the combined efforts of Baskett et al. [1975]. The result is known as the BCMP theorem, bearing their initials. Although this theorem is applicable to both open and closed networks, we present the result only in the context of closed queueing networks. Consider a closed queueing network with $N$ nodes. Each node has an infinite capacity; that is, there is always a space at a node for arriving customers. The topology of the network is arbitrary. There are $R$ different types of customer classes. A customer of class $r$, upon completing service at node $i$ goes to node $j$ with probability $p_{ij;r}$. $K_r$ is the number of class $r$ customers circulating in the network. Let $S$ and $S_i$ denote, respectively, the state of the network and the state of node $i$, that is, $S = (S_1, S_2, \ldots, S_N)$, whereas $S_i$ depends on the type of node $i$. In particular, node $i$ can be one of the following four types, referred to as the BCMP nodes.

### Type 1 Node

All customers have the same service time distribution that is exponentially distributed with mean $1/\mu_i$. Customers are served in order of arrival, referred to as the FCFS service discipline. The state $S_i$ of the node is defined as the vector $(r_1, r_2, \ldots, r_{n_i})$, where $n_i$ is the number of customers present at node $i$ and $r_j$ is the class index of the $j$ th customer in the FCFS order. There is a single server whose speed $C_i(n_i)$ depends on the number of customers at node $i$; that is, the instantaneous service completion rate at node $i$ is $\mu_i C_i(n_i)$.

### Type 2 Node

There is a single server and the service discipline is processor sharing (i.e., when there are $n$ customers at the node, each is receiving service at a rate of $1/n$ of the service rate). The service times for class $r$ customers are Coxian with parameters $a_{irl}$, $b_{irl}$, $\mu_{irl}$, and $L_{ir}$. The node state $S_i$ is defined as the vector $(v_1, v_2, \ldots, v_R)$, where $v_r = (n_{ir1}, n_{ir2}, \ldots, n_{irL_{ir}})$ is a vector whose $m$th element $n_{irm}$ denotes the number of class $r$ customers at node $i$ that are in the $m$th stage of their service. The speed of service at node $i$ may depend on the total number of customers at node $i$ as for type 1 nodes.

### Type 3 Nodes

There are as many servers as there may be customers requiring service. As soon as a customer arrives, a separate server is assigned for the duration of the customer's service. The assumptions regarding the required service time distributions and the definition of the node state $S_i$ are the same as for type 2 nodes.

### Type 4 Nodes

A single server is scheduled according to the preemptive resume *last come first served* (LCFS)discipline . In this discipline, a customer in service is preempted by an arriving customer. That is, when a new customer arrives the service of the current customer is interrupted until the new customer departs (which, in turn, may be interrupted) and then resumed from the point

of interruption. The service time is Coxian, as for type 2 and 3 nodes. The state of the node, $S_i$, is defined as the vector of pairs $\{(r_1, m_1), (r_2, m_2), \ldots, (r_{n_i}, m_{n_i})\}$, where $r_j$ and $m_j$ are, respectively, the class index and the stage of service of the $j$th customer in the LCFS order. The speed of the server may depend on the total number of customers at node $i$, as for type 1 nodes.

Let $M_r(S_i)$ be the number of class $r$ customers at node $i$ when the node is in state $S_i$. Then network state $S$ is feasible if $0 \le M_r(S_i) \le K_r$; $i = 1, \ldots, N$ and $\sum_{i=1}^{N} M_r(S_i) = K_r$; $r = 1, \ldots, R$. The steady-state queue length distribution $p(S)$ is the solution of the global balance equations together with the normalizing equation

$$p(S)\left[\begin{array}{r}\text{instantaneous transition rate}\\ \text{out of state } S\end{array}\right] \quad (1a)$$

$$= \sum_{S'} p(S')\left[\begin{array}{r}\text{instantaneous transition}\\ \text{rate from } S' \text{ to } S\end{array}\right]$$

$$\sum_{S} p(S) = 1 \quad (1b)$$

The *visit ratios*, $e_{ir}$, is defined as the *relative* arrival rate of class $r$ customers to node $i$. $e_{ir}$'s are determined from the following system of equations:

$$e_{ir} = \sum_{j=1}^{N} e_{jr} p_{ji;r}, \quad i = 1, \ldots, N; \quad r = 1, \ldots, R. \quad (2)$$

For each class $r$, there are $N$ unknowns ($e_{ir}$'s) and $N$ equations of which $N - 1$ are linearly independent. Hence, it is necessary to set one of the unknowns to an arbitrary value and calculate the others relative to the set value.

The following result is known as the BCMP theorem [Baskett et al. 1975]:

**Theorem 1**

*Let $e_{ir}$ ($i = 1, \ldots, N$; $r = 1, \ldots, R$) be any solution of* (2). *The general solution of the global balance equations* (1a) *has the form*

$$p(S) = G \prod_{i=1}^{N} f_i(S_i) \quad (3)$$

*where $G$ is the normalization constant that ensures that $\sum_S p(S) = 1$. The factor $f_i(S_i)$ depends on the type of node $i$:*

*If node $i$ is of type* 1, *then*

$$f_i(S_i) = \prod_{j=1}^{n_i} \left[\frac{e_{ir}}{\mu_i C_i(j)}\right].$$

*If node $i$ is of type* 2, *then*

$$f_i(S_i) = n_i! \frac{\prod_{r=1}^{R} \prod_{l=1}^{L_{ir}} \left[\left(\frac{e_{ir}A_{irl}}{\mu_{irl}}\right)^{n_{irl}} \Big/ n_{irl}!\right]}{\prod_{j=1}^{n_i} C_i(j).}$$

*If node $i$ is of type* 3, *then*

$$f_i(S_i) = \prod_{r=1}^{R} \prod_{l=1}^{L_{ir}} \left[\left(\frac{e_{ir}A_{irl}}{\mu_{irl}}\right)^{n_{irl}} \Big/ n_{irl}!\right].$$

*If node $i$ is of type* 4, *then*

$$f_i(S_i) = \prod_{j=1}^{n_i} \left[\frac{e_{ir_j}A_{ir_j|_j}}{\mu_{ir_j|_j} C_i(j)}\right].$$

Since the steady-state queue length distribution is the product of the functions of nodes, these types of solutions to the global balance equations are referred to as *product form solutions.*

Any Markovian model can, in theory, be solved numerically. In particular, obtaining the steady-state queue length distribution, $\pi$, of a queueing network is a three-step procedure: (1) Determine the states and the state space of the network; (2) determine its state transition structure to construct the rate matrix, $Q$, of the network; and (3) solve the linear system of equations $Q\pi = b$ numerically. There are, however, some practical limitations in obtaining the steady-state queue length distribution of queueing networks numerically. First, the state space of queueing networks grows rapidly with the number of nodes and the number of customers in the network. For example, the state space of a single class closed network with 10 type 1 nodes and 10 customers in it has 1,847,560 states. Hence, the space required to store the rate matrix $Q$ can be excessive; this problem, to

some extent, can be alleviated by using the sparseness of $Q$ (i.e., storing only the nonzero elements of $Q$). Second, the construction of $Q$ from a model is rather a time-consuming task. Finally, solving the system of equations, in general, has a time complexity of $o(n^3)$, where $n$ is the number of states, which restricts the applicability of numerical techniques. The interested reader may refer to Jennings [1977] for the details on solving systems of linear equations.

Simulations may be used to obtain the steady-state queue length distributions of queueing networks. Simulations could, however, be considered an approximation technique [Chandy and Sauer 1978]. In particular, the exact values of the steady-state queue length probabilities of a queueing network can be obtained if the network has a product form queue length distribution or a tractable numerical solution. However, the values obtained from simulation will, it is hoped, be near (but usually not equal to) the exact values. Two of the principal problems with simulations are determining how close simulation estimates are to the exact values and determining how long to run the simulation in order to obtain estimates near the correct values. Furthermore, developing and implementing simulation models are usually a time-consuming task. The interested reader may refer to Law and Kelton [1982] and Solomon [1983] for the details on modeling and analysis of queueing networks with simulation.

Approximate solution techniques have been developed to analyze queueing networks in which obtaining the exact solutions of the performance measures are inordinately expensive or the form of their steady-state queue length distributions are not known. The main problem with approximations is to bound the error in the solution. The accuracy of an approximation is tested with numerical solutions (in smaller configurations that can be solved numerically) or with simulations to determine the conditions under which the algorithm yields a good approximation. Decomposition methods are the most widely used techniques in the approximate analysis of queueing networks [Chandy and Sauer 1978; Chandy et al. 1975; Courtois 1977]. The main idea is to decompose the network into subnetworks, analyze each subnetwork in isolation, and use the results obtained from each subsystem to analyze the macro-system composed of these subsystems [Muntz 1978]. These methods give exact solutions for queueing networks with product form steady-state queue length distributions. In networks with nonproduct form solutions, the method yields good approximations if the rate of interaction among the nodes in the subnetwork is significantly higher than the rate of interaction of the subnetwork with the remainder of the network. The error bounds for this approach can be calculated from the rate matrix of the network [Courtois 1977].

Almost all queueing networks with product form queue length distributions require infinite queues; that is, it is assumed that there is always a space in the queue for arriving customers. In real life systems, the storage space is always finite. Hence, a more realistic model of such systems requires modeling finite node capacities. An important feature of queueing networks with finite queues is that the flow of customers through a node may be momentarily stopped when another node in the network reaches its capacity. That is, a phenomenon called *blocking* occurs. In particular, consider a simplistic view of a computer communications system. The individual queues represent the finite space that is available for intermediate storage and servers correspond to communication channels. A message may not be transmitted until the destination node has space available to store the message, thus sometimes causing the blocking of communication to that node. Similarly, in production systems, intermediate storage areas have finite capacities. A unit completing its service at a station may be forced to occupy the machine until there is a space available in the next station. While the unit blocks the machine, it may not be possible for the machine to process other units waiting in its queue. As another example, consider a multiprocessor system consisting of $N$ processors and $M$ memory modules connected

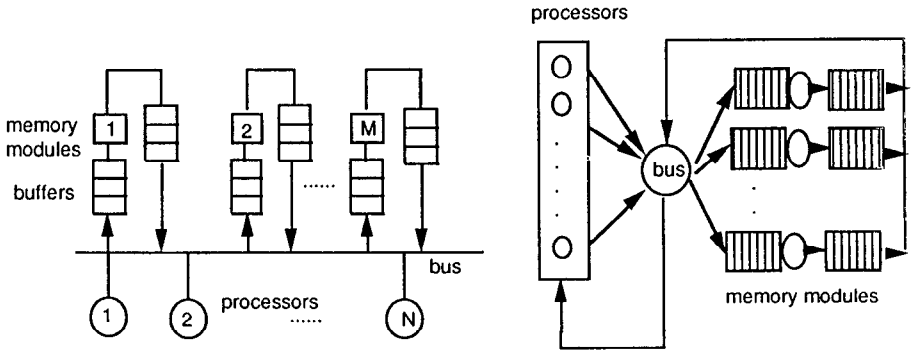


**Figure 7.** Multiprocessor architecture and its queueing model.

together by a multiplexed single bus. The memory modules have buffers at their inputs to queue the service requests of processors and buffers at their outputs to queue the requests served by the memory modules that cannot be served by the bus immediately. The system architecture and the queueing model of the system are illustrated in Figure 7.

Processor $i$ makes a request to memory module $j$. If the bus at that moment is not busy transferring a request for another processor or data from a memory module, processor $i$ takes the bus and the request is sent to memory module $j$. If the bus is busy transferring data, then processor $i$ has to retry its request at a later time. If the memory module $j$ is free, it will serve the request; if it is not free, the request will be queued. After the memory module completes its service, the output is placed in its output buffer for the bus to be transmitted to the processor that made the request. The effect of a full node on its upstream nodes (nodes that have a directed arc to the full node) depends on the type of system being modeled. If the input buffers of the memory modules are full, the bus cannot place the request to the buffer, and the processor has to send a new request. The request will be transmitted a number of times until it is delivered by the bus at a moment that there is a space at the buffer. Similarly, the output buffer of a memory module can be full. In this case, the module may be forced to suspend its service until a request is delivered from its output buffer to the processor that made the request, that is, until a space becomes available at the output buffer. Hence, distinct models for blocking have been reported in the literature to model various real life systems with finite resources.

In addition to the problem of blocking, *deadlocks* may occur in queueing networks with finite queues. In particular, a set of nodes is in a deadlock state when every node in the set is waiting for a space to become available at another node in the set. In this case, all servers in the set are blocked, and they can never get unblocked because the space required for the change of status will never be available. The problem of deadlocks is not unique to queueing networks with finite queues. Different aspects of the problem such as detection, avoidance, and prevention have been investigated in the literature, particularly in the context of operating systems [Coffman et al. 1971; Minoura 1982]. This problem, however, has been largely ignored in queueing networks with blocking. We will elaborate on this issue in Section 1.

This paper gives a survey of exact, approximate, and numerical results related to closed queueing networks with finite queues. The parameters describing these networks and distinct models of blocking that exist in the literature are defined in Section 1. In Section 2 we deal with two-node closed queueing networks. In Section 3, we survey results related to queueing networks with more than two nodes. We introduce the concept of indistinguishable

nodes in Section 4 and show that the steady-state queue length distribution of symmetrical networks can be obtained numerically on a reduced state space. Finally, we present conclusions in Section 5.

## 1. PRELIMINARIES

Queueing networks with blocking are difficult to solve; in general, their steady-state queue length distributions could not be shown to have product form solutions. Hence, most of the techniques used to analyze these networks are in the form of approximations, simulation, and numerical techniques. In recent years, there has been a growing interest in the development of computational methods for the analysis of both open and closed queueing networks with blocking. A comprehensive survey of the literature on open queueing networks with blocking was compiled by Perros [1989]. Although studies reported in the literature for open networks with blocking are of particular interest, these networks can be viewed as special types of closed networks. In particular, any open queueing network with finite queues and Poisson arrivals can be analyzed exactly as a closed queueing network (but not vice versa). Hence, the studies reported in the literature for closed queueing networks with blocking are directly applicable to open networks with finite queues [Onvural and Perros 1988].

Closed queueing networks considered here consist of $N$ nodes and $K$ customers. A customer which completes its service at node $i$ will next require service at node $j$ with a certain probability denoted $p_{ij}$. $B_i$ is the capacity (queue size plus one for the server) of node $i$; $i, j = 1, \ldots, N$. Throughout the paper, we assume that customers at each node are served in a FCFS manner and there is a single server with one stage of service at each node. These parameters are summarized in Table 1.

Consider a closed queueing network with parameters given in Table 1. Furthermore, assume that the service time at each node is exponentially distributed with mean $1/\mu_i$ and $B_i = \infty$, $i = 1, \ldots, N$. The state of this network, $S$, is an $N$-vector $(n_1, n_2, \ldots, n_N)$, where $n_i$ is the number of customers at node

**Table 1.** Parameters of Closed Queueing Networks

| | |
|---|---|
| $K$ | Number of customers in the network. |
| $N$ | Number of nodes. |
| $p_{ij}$ | Fraction of departures from node $i$ that proceed next to node $j$. |
| All nodes have FCFS service discipline. | |
| Each node has a single server with one stage of service. | |
| $B_i$ | Capacity of node $i$. |

$i$. We have $0 \le n_i \le B_i$ and $\sum_{i=1}^{N} n_i = K$. Then the steady-state queue length distribution of the network, $\pi(n_1, n_2, \ldots, n_N)$, has a product form solution as follows [Gordon and Newell 1967b]:

$$\pi(n_1, n_2, \ldots, n_N) = G \prod_{i=1}^{N} \left(\frac{e_i}{\mu_i}\right)^{n_i},$$

where $e_i$ is the relative visit ratio of node $i$ given by (2) with a single class of customers in the network (Equation 7 below) and $G$ is the normalization constant.

To understand and to predict the behavior of a real life system, an analyst would like to know the values of the performance metrics of the system such as the percentage of time a device is used, the rate at which the system produces an output, the average unfinished work at each device, average time it takes to produce a unit at each device, and average time it takes to produce a finished product. Corresponding to these, the primary performance measures of queueing networks are defined as follows: (i) marginal queue length distribution, (ii) use of each server, (iii) throughput of each node, (iv) throughput of the network, (v) mean queue lengths, and (vi) average response time at node $i$.

The steady-state *marginal queue length probabilities*, $p_i^K(n)$, of node $i$ is the steady-state probability of having $n$ customers at node $i$ when there are $K$ customers in the network, independent of the number of customers at other nodes of the network. Let $S$ denote a network state, $p(S)$ be the steady-state queue length probability of being in state $S$, and $Y_i(n)$ be the set of states such that there are $n$ customers at node $i$. Then

$$p_i^K(n) = \sum_{S \in Y_i(n)} p(S), \qquad n = 0, 1, \ldots, \min(K, B_i). \tag{4}$$

The *utilization*, $u_i(K)$, of node $i$ is defined as the percentage of time the server is busy serving when there are $K$ customers in the network. In the case of nonblocking networks (i.e., networks with infinite node capacities), this is equivalent to the probability that the node is not empty; that is, there is at least one customer at node $i$. In case of blocking networks, the existence of at least one customer at node $i$ does not necessarily mean that the server is busy serving as node $i$ may be blocked. The *effective utilization*, $u_i^E(K)$, of node $i$ is defined as the percentage of time the server is busy serving, whereas the *total utilization*, $u_i^T(K)$, of node $i$ is defined as the percentage of time there is at least one customer at node $i$. In terms of marginal queue length probabilities, we have

$$u_i^E(K) = (1 - p_i^K(0) - p_i^K(b)),$$

and (5)

$$u_i^T(K) = (1 - p_i^K(0))$$

where $p_i^K(b)$ is the probability that node $i$ is blocked.

The *throughput of node i* is defined as the rate at which customers depart from that node. Let $\lambda_i(K)$ be the throughput of node $i$ when there are $K$ customers in the network. Then,

$$\lambda_i(K) = \{1 - p_i^K(0) - p_i^K(b)\}\mu_i, \quad (6)$$

where $\mu_i$ is the service rate at node $i$. In single class networks, the equations for the visit ratios, $e_i$, defined in (2) reduces to the following:

$$e_i = \sum_{j=1}^{N} e_i p_{ji}, \quad i = 1, \ldots, N. \quad (7)$$

We note that there are $N$ unknowns and $N - 1$ independent equations in the above system. Hence, it is necessary to set one of the $e_i$'s to one (or any other value) and solve the other $e_i$'s relative to that given value. Without loss of generality, let the visit ratio of node 1 be set to one. In this case, the *throughput of the network*, $\lambda(K)$, is defined as the average number of customers leaving node 1 per unit time. Furthermore, the relation between the throughputs of nodes and the throughput of the network is given as follows:

$$\lambda_i(K) = e_i\lambda(K), \quad i = 1, \ldots, N. \quad (8)$$

The *mean queue length*, $L_i$, of node $i$ is the average number of customers in node $i$ at steady state; that is,

$$\sum_{n=1}^{\min(B_i,K)} p_i^K(n). \quad (9)$$

The *average response time*, $w_i$, is the average time a customer spends in node $i$ at steady state. There is a simple relation between the average response time and the mean queue length in a queueing system in steady state that equates the average arrival rate to the average departure rate. In particular, since a customer remains at node $i$ for an average time of $w_i$, the departure rate is $1/w_i$. The average number of customers at node $i$ is $L_i$. Hence, the average departure rate is $L_i/w_i$. In steady state, the average arrival rate to node $i$, $\lambda_i$, is equal to the average departure rate from node $i$; hence, we have the following result, which is known as the *Little's result* [Little 1961]:

$$L_i = \lambda_i w_i. \quad (10)$$

### 1.1 Blocking Mechanisms

The effect of a full node on its downstream nodes depends on the type of the system being modeled. For presentation purposes, consider the simplistic view of a transaction processing (TP) system as shown in Figure 8 [Highleyman 1989].

Users of the system send their requests (read or update) to a request handler. When the request handler receives a request from a user, it evaluates the request and passes it to an appropriate server that is designed to handle that request type. Figure 8, being very simplistic, illustrates two types of servers: one for handling inquiries and one for handling updates. The servers usually interact with the database manager to gain access to (or to update) data in the database. It then formulates a reply and returns it to the user that made the request via a reply handler.

A queueing model of this transaction processing system is shown in Figure 9.

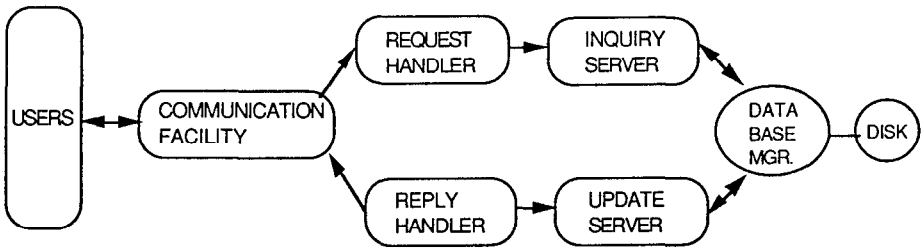


**Figure 8.** Transaction processing system.

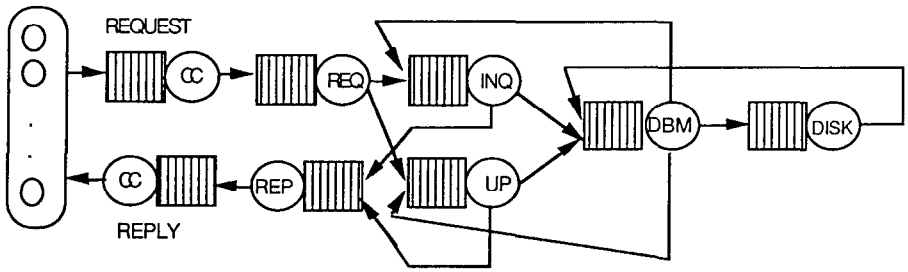


**Figure 9.** Queueing model of a TP system.

User requests are transmitted from user terminals to request handler (REQ) over a communication channel (CC). Once the request is received and processed by the request handler, it is passed to the queue of the appropriate server: inquiry server (INQ) or update server (UP). The server processes the request and sends it to the database manager (DBM), which accesses the disk to serve the request. Once the request is served by the disk, the data (for inquiry operation) or status (for update operation) is returned to the appropriate server. When the server completes its processing, it sends a reply to the reply handler (REP), which in turn returns the reply to the user via the communication link. The queues between various servers represent the buffers available for intermediate storage. Since there is a finite number of buffers available at each node, it is possible that one or more of the queues are full at any given time. In particular, a user generates a request and attempts to access the communication channel. If the channel is busy transferring another request and if there is a buffer available, then the request is queued. If, however, there is no buffer available, then the user suspends its operation; that is, it cannot generate new requests.

Similarly, the communication to the request handler may be temporarily stopped if there is no space available in the request handler queue. Furthermore, the communication channel cannot be used to store a request due to physical constraints; hence all requests should wait in the queue until there is a space available in the request handler queue at which time the communication may be resumed. Request handler upon completing the processing of a request attempts to place the request at appropriate server's queue. If there is no space available at the destination node, the request handler suspends its service until a space becomes available at the destination node. Finally, the database manager may have to send the request to the disk a number of times before a space becomes available at the disk buffers. Hence, the effect of a full node on the service of the nodes that are connected to it may be quite different depending on the

type of system being modeled. In particular, a full node may not affect the operation of an upstream server until the service is completed (user request sent to the communication channel), or the server may not start serving its customer until there is a space available in the destination node (communication channel may not transmit if there is no space available in the request handler queue). Service may be suspended during the period that the destination node is full, or a customer may be served a number of times until it is accepted by the destination node (database manager sends a request a number of times until there is a space available in the disk buffers).

To model different characteristics of various real life systems with finite resources, various blocking mechanisms that define distinct models of blocking have been reported in the literature. In particular, each blocking mechanism defines when a node is blocked, what happens during the blocking period, and how a node becomes unblocked. Following Onvural and Perros [1986] and Perros [1989], we next classify the most commonly used blocking mechanisms.

### *1.1.1 Blocked after Service (BAS)*

A customer upon completion of its service at node $i$ attempts to enter destination node $j$. If node $j$ at that moment is full, the customer is forced to occupy server $i$ until it enters destination node $j$, and node $i$ is blocked. Node $i$ remains blocked for this period of time, and server $i$ cannot serve any other customer that might be waiting in its queue. In queueing networks with arbitrary topologies, it is possible that a number of nodes may be blocked by the same node simultaneously. This necessitates imposing an ordering on the blocked nodes to determine which node will be unblocked first when a departure occurs from the blocking node. This problem has not been elaborated on in the literature. We are only aware of the *First-Blocked-First-Unblocked* rule (FBFU), which states that the node that was blocked first will be unblocked first [Altiok and Perros 1987].

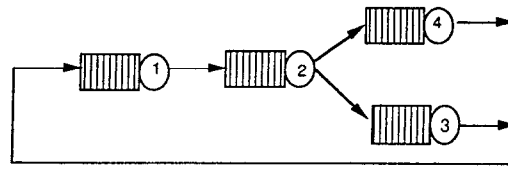


**Figure 10.** Four-node closed queueing network.

Consider the four-node network given in Figure 10 with $B_i = 2$, $i = 1, \ldots, 4$; $K = 6$ and assume that the network is in a state where there are two customers each in nodes 1, 3, and 4; all servers are busy serving. Let server 4 complete its service before servers 1 and 3 blocking node 4. If server 3 completes its service before server 1, then both nodes 4 and 3 are blocked by node 1, in that order. Upon service completion, customer at node 1 goes to node 2. At that moment, blocked customer at node 4 will join node 1, unblocking node 4, while node 3 will have to wait for another service completion at node 1 before it is unblocked.

It is possible that deadlocks might occur in queueing networks under BAS blocking. Let us consider the above example and assume that the network is in a state in which node 1 is blocked by node 2, node 3 is blocked by node 1, and node 3 is full (i.e., there are two customers each in nodes 1, 2, and 3 and no customer at node 4). If the customer at node 2, upon service completion, chooses to go to node 3, then a deadlock occurs. The problem of deadlocks in networks under BAS blocking has been addressed in Onvural and Perros [1989b], Jun [1988], and Akyildiz and Kundu [1989]. In particular, it was assumed in the first two references that such deadlocks can be detected immediately and resolved by instantaneously exchanging blocked customers; that is, in the above example, when a deadlock occurs blocked customers at nodes 1, 2, and 3 simultaneously join nodes 2, 3, and 1, respectively. Lemma 1 gives the necessary and sufficient condition for a network under BAS blocking to be deadlock free [Kundu and Akyildiz 1989]. A *cycle* of a network is defined as a directed path that starts and ends at the same node. For example, in the four-node closed network

illustrated in Figure 10, the two cycles that start and end at node 1 are (1, 2, 3, 1) and (1, 2, 4, 1).

**Lemma 1**

*A closed queueing network under BAS blocking 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 the node capacities in each cycle.

The BAS blocking (also referred to as type 1 blocking, manufacturing blocking, classical blocking, and transfer blocking) has been used to model systems such as production systems and disk I/O subsystems [Altiok and Perros 1987; Perros 1984, 1989; Suri and Diehl 1984]. In particular, consider a simplistic view of a production system having a sequence of operation stations. The availability of a storage space in the next destination has no effect on the operation of a machine until it completes its service. Upon service completion, if there is no space available in the destination station, then there are two possibilities: The unit that completed its service at node $i$ is (1) moved back from the $i$th machine to the storage area of machine $i$ (i.e., its queue) allowing a unit to enter service or (2) allowed to occupy the machine, blocking the operation for other units waiting in the storage space (i.e., BAS blocking). In most cases, it may not be possible to move the unit from the machine back to the storage area due to the physical constraints of the unit, the system, or both. Hence, the operation of the machine is blocked until there is a space available at the destination station. In a recent paper, Mitra and Mitrani [1988] considered the possibility of moving the blocked customer back to its queue in the context of open networks to model the Japanese Kenban scheme that is used for cell coordination in production lines. The interested reader may refer to Mitra and Mitrani [1988] for a detailed explanation of this scheme.

*1.1.2 Blocked before Service (BBS)*

A customer at node $i$ declares its destination node $j$ before it starts receiving its service. If node $j$ is full, the $i$th node becomes blocked. When a departure occurs from the destination node $j$, node $i$ is unblocked and its server starts serving the customer. If the destination node $j$ becomes full during the service of a customer at node $i$, the service is interrupted and node $i$ is blocked. The service is resumed from the interruption point as soon as a space becomes available at the destination node. Depending upon whether the customer is allowed to occupy the service area when the server is blocked, the following subcategories are distinguished.

**BBS-SNO** (*s*erver is *n*ot *o*ccupied) Service facility of a blocked node cannot be used to hold a customer.

**BBS-SO** (*s*erver is *o*ccupied) Service facility of a blocked node is used to hold a customer.

In BBS blocking mechanism, a full node $j$ blocks all nodes $i$ that are connected to it (i.e., $p_{ij} > 0$). When a departure occurs from node $j$, all blocked nodes become unblocked simultaneously and start serving their customers. Hence, there is no need to impose any ordering on the blocked nodes, unlike BAS blocking.

The BBS blocking mechanism (also called type 2 blocking, immediate blocking, service blocking, communications blocking) is motivated by considering servers that only move customers between stations and do no other work on them. In this case, the lack of downstream space must force the server to shut down. Two nodes $i$ and $j$ are called adjacent if there is a directed arc that connects node $i$ to node $j$, that is, $p_{ij} > 0$, and $B_i + B_j$ (the capacity of two adjacent nodes) is the upper limit on the number of customers that can be accommodated simultaneously in nodes $i$ and $j$. Furthermore, let $K'$ be the number of customers in the network such that there can be only one node blocked at a time and the blocked node cannot be full. Then, $K' = \min\{B_i + B_j;\ i, j = 1, \ldots, N \text{ s.t. } p_{ij} > 0\}$.

We note that there is no difference between the BBS-SO and BBS-SNO blocking mechanisms when only one node can be blocked at a time and the blocked node cannot be full (Lemma 2b). The two blocking mechanisms, however, are not equivalent when $K > K'$. Consider an $N$ node ($N > 2$) cyclic network with $K = K' + 1$. In BBS-SO blocking, it is possible to have $B_i$ and $B_{i+1}$ customers at nodes $i$ and $i + 1$, respectively, such that $K = B_i + B_{i+1}$. In BBS-SNO blocking, however, there can be at most $B_i - 1$ customers at node $i$ when node $i + 1$ is full. Hence, the set of global balance equations that describes the stochastic behavior of a cyclic network (i.e., its rate matrix) is different under the two blocking mechanisms. The above discussion may suggest that there might be an equivalence between the two blocking mechanisms after the node capacities are increased by 1 in BBS-SNO blocking. In this case, however, the number of ways that $K$ customers can be distributed over $N$ nodes is not the same for the two blocking mechanisms. Hence, in general, BBS-SO and BBS-SNO blocking mechanisms have different stochastic behavior and there is no equivalence between the two.

The distinction between BBS-SO and BBS-SNO blocking mechanisms is meaningful when modeling different types of systems. For example, in communication networks, a server corresponds to a communication channel. If there is no space in the downstream node, then messages cannot be transmitted. Furthermore, the channel itself cannot be used to store messages due to physical constraints of the channel; that is, BBS-SNO blocking. On the other hand, BBS-SO blocking results if the service facility can be used to hold the blocked customer, which, in this case, would be an approximate modeling of the system.

BBS-SO blocking has been used to model manufacturing systems, terminal concentrators, mass storage systems, disk-to-tape back-up systems, window flow control mechanisms, and communication systems [Ammar and Gershwin 1987; Boxma and Konheim 1981; Gershwin and Berman 1981], all in the context of open networks and Suri and Diehl [1986]. Modeling these systems with BBS-SO blocking assumes that when its destination buffer is full the device is forced to stop its operation and the service facility can be used to hold a customer. A disk-to-tape back-up model illustrated in Figure 11 is comprised of three servers and two finite buffers between servers. The first server is the disk and channel that transfers blocks of data from the disk to the main memory. The second server, the CPU, transfers data from the main memory to the tape drive. The last server represents the tape drive. One of the performance objectives of interest is the tape back-up rate (i.e., the throughput of the system). Blocking occurs due to finite spaces available for intermediate storage.

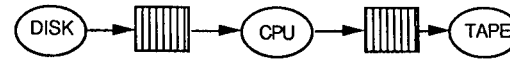


**Figure 11.** Disk-to-tape backup system.

The next example is motivated by a simple mass storage system (MSS) as might be used in a data processing environment. The system consists of a first MSS, a staging disk, a CPU, an outstaging disk, and another MSS, as illustrated in Figure 12. Due to the relatively small sizes of the buffers, the blocking primarily occurs between the disks and the MSS devices.

A simple terminal concentrator consists of a number of terminals, a concentrator, and a channel to transfer data to the main memory. The system configuration is the same as the disk-to-tape back-up system illustrated in Figure 11, with the concentrator, the channel, and the CPU replacing the disk, the CPU, and the tape, respectively. Similar to the above two examples, the two buffers in this terminal concentrator system have finite capacities that cause blocking of respective nodes. We note that the above examples are only subsystems of larger configurations of computer systems, used only to illustrate the possibility of blocking due to finite storage capacities between the devices of such systems.

For example, consider the disk-to-tape back-up system. The first server corresponds to both the disk and the channel. If there is no space available in the memory,

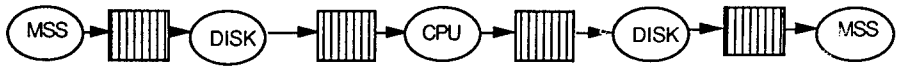


**Figure 12.** Mass storage system.

then the transfer of data has to be suspended. The server will resume its operation when a space becomes available at the memory. Similarly, other servers are forced to suspend their services if there is no space available at their destination nodes.

Closed queueing networks under BBS blocking are not always well defined for arbitrary topologies with an arbitrary number of customers in the network. This is because deadlocks in this blocking mechanism cannot be resolved without violating its rules. As an example, let us assume that node $i$ is blocked by node $j$ and node $j$ is blocked by node $i$. Then the services at both nodes are suspended. Furthermore, the service cannot start unless the blocking mechanism is temporarily switched to, for example, BAS blocking. In view of this, this blocking mechanism can only be used in deadlock free networks. Similar to Lemma 1, it can be shown that a closed queueing network with BBS blocking is deadlock free if and only if for each cycle $C$ in the network, (i) $K < \sum_{j \in C} \{B_j - 1\}$ in BBS-SNO blocking, and (ii) $K < \sum_{j \in C} B_j$ in BBS-SO blocking. Simply stated, a closed network under BBS-SO blocking is deadlock free if the number of customers in the network is less than the sum of node capacities in each cycle in the network, whereas in a BBS-SNO, a network is deadlock free if the number of customers in the network is less than the sum of queue capacities (node capacity minus 1 for the server facility) in each cycle.

**Repetitive Service** (RS) A customer upon service completion at node $i$ attempts to join destination node $j$. If node $j$ at that moment is full, the customer receives another service at node $i$. This is repeated until the customer completes a service at node $i$ at a moment that the destination node is not full. Within this category of blocking mechanisms, the following two subcategories are distinguished:

RS-FD ($f$ixed $d$estination) Once the customer's destination is determined, it cannot be altered.

RS-RD ($r$andom $d$estination) A destination node is chosen at each service completion independently of the destination node chosen the previous time.

We note that closed queueing networks with exponential servers have the same rate matrix under both BBS-SO and RS-FD blocking mechanisms (Lemma 2a). Hence, a network under RS-FD blocking is deadlock free if and only if $K < \sum_{j \in C} B_j$ for each cycle in the network; that is, the number of customers in the network is less than the capacity of each cycle in the network. On the other hand, it can be shown that a closed network under RS-RD blocking is deadlock free if the network is irreducible (i.e., there is a path from every node to every other node in the network) and if there is at least one free space in the network; that is, $K < \sum_{i=1}^{N} B_i$ (not for each cycle). This is because the existence of a free space in the network guarantees that all blocked customers will eventually depart, unblocking their servers.

The RS blocking (also called rejection blocking and type 3 blocking) arise in modeling telecommunication systems and is mostly associated with reversible queueing networks. In particular, let us consider a packet switching network with fixed routing. The number of packets in the network is controlled by a window flow mechanism [Reiser 1979]. A node transmits a packet to a destination node and waits for an acknowledgment. If the destination node does not accept the packet due to the fact that there is a lack of space, it will not send an acknowledgment. In this case, the packet may be retransmitted (RS-FD blocking) until it is accepted by the destination node (i.e., until an acknowledgment is received

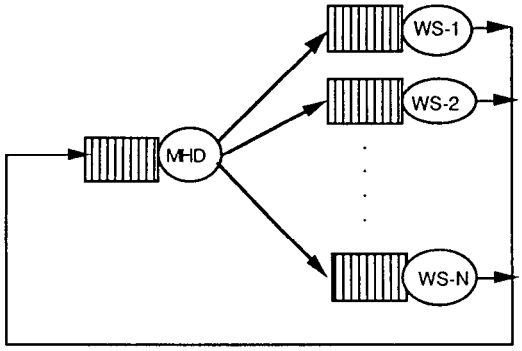


**Figure 13.** Queueing model of a flexible manufacturing system. MHD, Material handling device; WS-$i$, $i$th workstation.

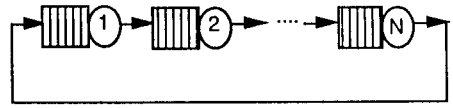


**Figure 14.** Cyclic network.

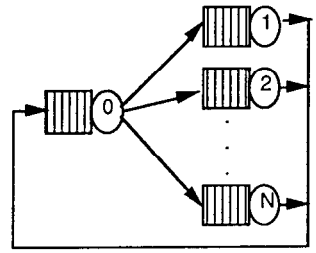


**Figure 15.** Central server model.

by the sender). Similarly, consider a manufacturing system consisting of a network of automated work stations (WS) linked by a computer controlled material handling device (MHD) to transport work-pieces that are to be processed from one station to another, as illustrated in Figure 13.

In these systems, if a work-piece finds the next station full, it has to wait for the next turn of the MHD. At the next turn, there are two possibilities: (i) the work-piece can only be processed by one station, therefore, the next attempt can only be made to the previously chosen station (i.e., RS-FD blocking), or (ii) the unit may be processed by all stations, hence, the next station is chosen independent of the previous choice(s) (i.e., RS-RD blocking). If the service time of the MHD is assumed to be exponentially distributed, the RS-RD blocking is equivalent to the following: The work-piece attempts to enter station 1. If station 1 is full, it tries station 2, and so on, until a space is found in one of the stations. The interested reader may refer to Yao and Buzacott [1985a, b, c, 1986] for the details on the flexible manufacturing systems.

### 1.2 Equivalences of Blocking Mechanisms

Comparisons between these distinct types of blocking mechanisms in the context of closed queueing networks have been carried out by Balsamo et al. [1986] and Onvural [1987]. The objective of these comparisons was to obtain an equivalence between different blocking mechanisms applied to the same 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 equivalences obtained in the literature assume that the service times are exponentially distributed. Furthermore, these equivalences are most often true only for specific topologies: cyclic networks and the central server model shown in Figures 14 and 15, respectively. In the central server model, there is a single node, referred to as the central server, connected to $N$ nodes. A customer upon completion of its service at the central server joins node $i$ with probability $p_{0i}$, $\sum_{i=1}^{N} p_{0i} = 1$, whereas customers at the other nodes join the central server with probability 1. A cyclic network consists of $N$ nodes connected in series. A customer upon completion of its service at node $i$ always joins the proceeding node $i + 1$. Customers at node $N$ always join the first node, forming a cycle. The following lemmas were proved Onvural [1987] and Balsamo et al. [1986].

**Lemma 2**

*In closed queueing networks with arbitrary topologies:*

(a) *BBS-SO and RS-FD are equivalent.*
(b) *BBS-SO and BBS-SNO are equivalent for*

$$K \leq \min\{B_i + B_j; \quad i, j = 1, \ldots, N \text{ s.t. } p_{ij} > 0\} - 1.$$

**Lemma 3**

*In cyclic networks*

(a) *RS-FD and RS-RD are equivalent.*
(b) *BBS-SNO given node capacities $B_i$ is equivalent to BAS blocking with node capacities $B_i - 1$, $i = 1, \ldots, N$.*

**Lemma 4**

*In the central server model,*

(a) *BBS-SO and BBS-SNO are equivalent if $B_1 = \infty$.*
(b) *BBS-SO and BBS-SNO are equivalent if $B_i = \infty$, $i = 2, \ldots, N$.*
(c) *RS-FD and RS-RD are equivalent if $B_i = \infty$, $i = 2, \ldots, N$.*

**Lemma 5**

*In cyclic networks with two nodes,*

(a) *BBS-SNO, BBS-SO, RS-FD, and RS-RD are equivalent.*
(b) *BBS-SNO (BBS-SO, RS-FD, and RS-RD) with node capacities $B_1$ and $B_2$ is equivalent to BAS with node capacities $B_1 - 1$ and $B_2 - 1$.*

## 2. TWO-NODE CLOSED QUEUEING NETWORKS

We start our review of the literature with two-node closed queueing networks illustrated in Figure 16. Let $K$ be the number of customers in the network and $B_i$ be the capacity of node $i$. Furthermore, assume that the service time at each node is exponentially distributed with rate $\mu_i$, $i = 1, 2$.

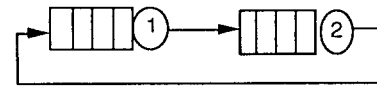


**Figure 16.** Two-node closed queueing network.

### 2.1 Blocked after Service Blocking

Two-node closed queueing networks under BAS blocking have been studied by Diehl [1984] and Akyildiz [1987]. In particular, Akyildiz demonstrated that a two-node closed network with $K$ customers under BAS blocking is equivalent (i.e., has the same rate matrix) to a nonblocking network, with the same parameters as the blocking network but with infinite capacities and with $K'$ customers, where

$$K' = \min(K, B_1 + 1) + \min(K, B_2 + 1) - K. \tag{11}$$

The state of this two-node network is defined as $(n_1, n_2)$, where $n_i$ is the number of customers at node $i$, $i = 1, 2$. Furthermore, the states $(k_1, B_2 + 1)$ and $(B_1 + 1, K - B_1)$ denote that nodes 1 and 2 are blocked, respectively. We note that in this blocking mechanism there is a need to distinguish the state $(k_1, B_2)$, where both servers are busy serving, from the state $(k_1, B_2 + 1)$, where node 1 is blocked, although the number of customers at each node is the same in both states. Let $k_1$ and $k_2$ be the minimum and maximum occupancy of node 1; that is, $k_1 = \max(0, K - B_2)$ and $k_2 = \min(K, B_1)$. If $K > \max(B_1, B_2)$, we have $k_1 > 0$ and $k_2 = B_1$. Then the rate diagram associated with this two-node network is given in Figure 17.

The equivalent nonblocking network has the same rate diagram with $K'$ customers and $B_1 = B_2 = K'$. Let $p(i, j)$ be the steady-state joint queue length probability of being in state $(i, j)$. The global balance equations can be written easily from the transition

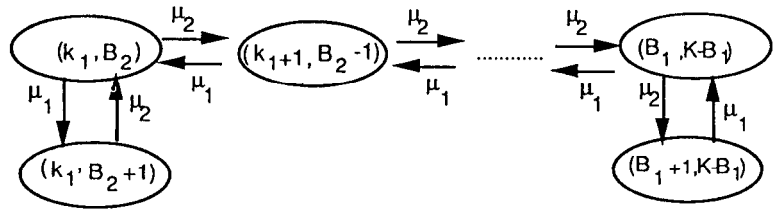


**Figure 17.** Transition rate diagram of a two-node network under BAS blocking.

rate diagram:

$$\mu_2 p(k_1, B_2+1) = \mu_1 p(k_1, B_2)$$

$$(\mu_2+\mu_1)p(k_1+i, B_2-i) = \mu_2 p(k_1+i-1, B_2-i+1) + \mu_1 p(k_1+i+1, B_2-i-1)$$

$$\mu_1 p(B_1+1, K-B_1) = \mu_2 p(B_1, K-B_1)$$

Solving the global balance equation for the steady-state queue length distributions $p(i, j)$, we have

$$p(k_1+i, B_2-i) = \left(\frac{\mu_2}{\mu_1}\right)^{i+1} p(k_1, B_2+1); \quad i = 0, \ldots, B_1 - k_1 + 1, \quad (12)$$

and $p(k_1, B_2+1)$ can be determined from the normalizing equation; that is,

$$p(k_1, B_2+1) \sum_{i=0}^{B_1-k_1+2} \left(\frac{\mu_2}{\mu_1}\right)^i = 1. \quad (13)$$

If $K \leq B_2$, then the first server can never get blocked since the capacity of the second node is large enough to hold all $K$ customers. In this case, the first node becomes identical to an M/M/1/$B_1$ + 1 queue with arrival and service rates equal to $\mu_1$ and $\mu_2$, respectively. If the node capacities of both nodes are large enough to hold all $K$ customers, then there is no blocking; hence, the network has a product form queue length distribution [Gordon and Newell 1967b].

Furthermore, these results are readily applicable to a two-node closed queueing network under BBS-SNO blocking after the node capacities increased by 1 (Lemma 3b).

### 2.2 Blocked before Service-Server Occupied Blocking

Now, consider the two-node closed queueing network under BBS-SO blocking, and let $K < B_1 + B_2$ to eliminate the problem of deadlock. This model was first studied by Gordon and Newell [1967a]. Following their argument, let $p(n_1, n_2)$ be the steady-state probability that there are $n_i$ customers at node $i$, $i = 1, 2$. Since a node is blocked and service is suspended when its destination node is full, it is not necessary to define additional states to define the states in which a node is blocked, unlike BAS blocking. Let $k_1$ and $k_2$ be the minimum and maximum occupancy in node 1, respectively, as defined above. Assuming $K \geq \max(B_1, B_2)$, the rate diagram associated with the network is illustrated in Figure 18.

The global balance equations are given as follows:

$$\mu_2 p(k_1, B_2) = \mu_1 p(k_1+1, B_2-1)$$

$$(\mu_2+\mu_1)p(k_1+i, K-k_1-i) = \mu_2 p(k_1+i-1, K-i-k_1+1) + \mu_1 p(k_1+i+1, K-i-k_1-1)$$

$$\mu_1 p(k_2, K-B_1) = \mu_2 p(k_2-1, K-k_2+1)$$

Solving the global balance equations, we have

$$p(k_1+i, K-k_1-i) = \left(\frac{\mu_2}{\mu_1}\right)^i p(k_1, B_2); \quad i = 1, \ldots, k_2 - k_1, \quad (14)$$

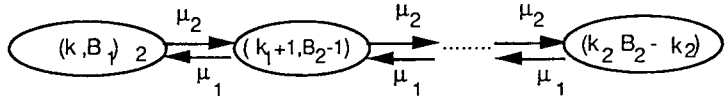


**Figure 18.** Transition rate diagram of a two-node network under BBS-SO blocking.

and $p(k_1, B_2)$ can easily be obtained from the normalizing equation; that is,

$$p(k_1, B_2) \sum_{i=0}^{k_2-k_1} \left(\frac{\mu_2}{\mu_1}\right)^i = 1 \qquad (15)$$

If $B_1 \leq K \leq B_2$, then the queue length distribution of the first queue becomes identical to an M/M/1/$B_1$/FCFS queue with arrival and service rates equal to $\mu_2$ and $\mu_1$, respectively. If both node capacities are greater than or equal to the number of customers, there is no blocking and the network has a product form steady-state queue length distribution [Gordon and Newell 1967b]. Furthermore, it can be easily shown that a two-node closed network under BBS-SO blocking has the same rate matrix as a nonblocking network with $K'$ customers, where

$$K' = \min(K, B_1) + \min(K, B_2) - K. \qquad (16)$$

We note that BBS-SO blocking is equivalent to RS-FD and RS-RD blocking mechanisms in two-node closed queueing networks (Lemmas 2 and 3a). Thus, the product form of the queue length distribution presented above for BBS-SO blocking is also applicable to the other two types of blocking mechanisms.

### 2.3 Two-Node Networks with Multiple Classes

The product form queue length distributions of two-node exponential blocking networks presented above can be extended to include multiple classes of customers and BCMP [Baskett et al. 1975] type of nodes by viewing a two-node blocking network as a truncated process of the same network with infinite node capacities. In particular, a process $T$ is called a *truncated process* of $Z$ [Kelly 1979] if (a) it is irreducible; (b) the state space of $T$ is a subset of the state space of $Z$, and (c) the transitions between the states of $T$ is the same as it is in $Z$. The routing in a network is called *reversible* if there exists positive $\lambda_i$ such that

$$p_{ij}\lambda_i = p_{ji}\lambda_j; \quad i, j = 1, \ldots, N. \qquad (17)$$

Equation (17) states that the rate at which customers arrive at node $j$ from node $i$ is equal to the rate at which customers leave node $j$ to return to node $i$. The following lemma is given in Kelly [1979]:

#### Lemma 6

*The form of the queue length distribution, $\pi(n)$, of a truncated process is the same as the original process normalized over the state space A of the truncated process; that is, $\sum_{n \in A} \pi(n) = 1$.*

A two-node network with buffer capacities $B_1$ and $B_2$ under BAS blocking is a truncated process in which no more than $B_i + 1$ customers are allowed at node $i$. Similarly, a two-node network under BBS-SO blocking is a truncated process in which no more than $B_i$ customers are allowed at node $i$, $i = 1, 2$. The following lemma, proved in Onvural [1989b], extends the set of networks with product form queue length distributions to include multiple classes and BCMP type of nodes (see also Van Dijk and Tijms [1986]).

#### Lemma 7

*A two-node closed network with multiple classes and BCMP type nodes has a product form queue length distribution under the blocking mechanisms defined in Section 1. Furthermore, the queue length distribution of the blocking network is the same as the queue length distribution of the network with infinite queue capacities normalized over the states of the blocking network.*

We now proceed with the survey of closed queueing networks consisting of more than two nodes.

## 3. CLOSED QUEUEING NETWORKS WITH MORE THAN TWO NODES

In this section, we survey results related to closed queueing networks with arbitrary topologies under the blocking mechanisms defined in Section 1.

### 3.1 Blocked after Service Blocking

Earlier work in queueing networks with finite queues was motivated by open queueing networks under BAS blocking in the context of production systems. Most of the approximations developed for these networks are based on decomposing the network into individual queues and analyzing them in isolation. Hence, these algorithms produce the marginal queue length probabilities from which other performance measures are calculated. On the other hand, there are relatively few results reported in the literature for closed queueing networks under BAS blocking, and much of the work has been done in recent years. There is no algorithm reported in the literature to approximate the marginal queue length probabilities of these networks.

Consider a closed queueing network with parameters given in Table 1 under BAS blocking. Furthermore, assume that the service time at each node is exponentially distributed with rate $\mu_i$, $i = 1, \ldots, N$. For $1 \leq K \leq \min B_i$, there is no blocking and the network has a product form queue length distribution [Gordon and Newell 1967b]. When $K \geq \min B_i + 1$ blocking occurs. In this case, product form queue length distributions are, in general, not available. When $K = \min(B_i, i = 1, \ldots, N) + 1$, however, there can be at most one node blocked at a time, and when a server is blocked there cannot be any customer waiting in its queue. Thus, during the blocking period, the service area in the blocked node behaves like an additional space for the blocking node. Onvural and Perros [1989a] showed that such networks, in this special case, have product form queue length distributions.

**Table 2.** States of the Central Server Model with $B_i = 2$; $i = 0, 1, 2$

| |
|---|
| No node is blocked: |
| {0, 1, 2} {0, 2, 1} {1, 0, 2} {1, 1, 1} {1, 2, 0} {2, 0, 1} {2, 1, 0}. |
| Node 1 is blocked by node 0: |
| $\{2, (1, 0)_1, 0\}$. |
| Node 2 is blocked by node 0: |
| $\{2, 0, (1, 0)_2\}$. |
| Node 0 is blocked by node 1: |
| $\{(1, 1)_0, 2, 0\}$. |
| Node 0 is blocked by node 1: |
| $\{(1, 2)_0, 0, 2\}$. |

**Lemma 8**

*Consider a closed exponential queueing network under BAS blocking with parameters given in Table* 1. *If the number of customers in the network, K, is equal to the minimum node capacity plus* 1, *then the network has a product form queue length distribution.*

To see this, let $(i, s)_k$ be the state of node $k$, where $i$ is the number of customers at node $k$ and $s$ is the index of the blocking node. If node $i$ is not blocked, then $s$ is dropped from the state definition. We have $0 \leq i \leq \min(B_k, K)$. For example, consider the central server model shown in Figure 15, with $N = 2$, $B_i = 2$; $i = 0, 1, 2$ with $K = 3$ customers. Then, the states of the network are given in Table 2.

Furthermore, let $p(S) = p((i, s)_1, (i, s)_2, \ldots, (i, s)_N)$ be the steady-state queue length probabilities of a closed network under BAS blocking. For a moment, assume that each node has an infinite capacity and let $\pi(j_1, j_2, \ldots, j_N)$ be its steady-state queue length probabilities, where $j_k$ is the number of customers at node $k$, $0 \leq j_k \leq K$, $k = 1, \ldots, N$. We note that $\pi(j_1, j_2, \ldots, j_N)$ has a product form queue length distribution. If the above assumptions are satisfied, then

$$p(S) = \begin{cases} \pi(i_1, i_2, \ldots, i_N) \\ \quad \text{if no node is blocked} \\ \dfrac{p_{mj} e_m}{e_j} \pi(0, \ldots, i_j \\ \qquad = B_j + 1, 0, \ldots, 0) \\ \quad \text{if node } m \text{ is blocked by node } j \end{cases} \tag{18}$$

where $e_i$ is the relative number of visits a customer makes to the $i$th node and is given by Equation (7).

Although Lemma 8 is a special case, it was used in Onvural [1987] and Onvural and Perros [1988] to obtain a lower bound on the throughput of closed and open networks, respectively. It can also be used in the validation process for approximations. Finally, we note that closed queueing networks with more than two nodes and under BAS blocking could not be shown to have product form queue length distributions other than this special case.

### 3.3.1 Throughput

Let $\lambda_i(K)$ and $\lambda(K)$ be, respectively, the throughput of a node $i$ and the throughput of the network when there are $K$ customers in the network. By definition, $\lambda_i(K) = \{1 - P_i^K(0), - P_i^K(b)\}\mu_i$, where $\mu_i$ is the service rate, $P_i^K(0)$ 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. Furthermore, we have $\lambda(K) = \lambda_1(K)$ [Equations (7) and (8) in Section 1]. Clearly, $\lambda(K)$ depends on the parameters of the network. Let $M$ be the capacity of the network (sum of node capacities); that is, $M = \sum_{i=1}^{N} B_i$. In Figure 19, we give an example of $\lambda(K)$ as $K$ changes from 1 to $M$ for the cyclic network shown in Figure 14 with $N = 3$ and $M = \sum_{i=1}^{3} B_i$. We note that in Figure 19, $\lambda(K)$ increases as $K$ increases until it reaches a maximum value, $\lambda^*$, for some $K^*$. For $K \geq K^*$, $\lambda(K)$ is nonincreasing (see also Persone and Grillo [1987]).

Lemma 9 provides bounds on the maximum throughput and the number of customers, $K^*$, that produces the maximum throughput [Onvural 1987], which is based on the following three conjectures:

**Conjecture 1**

*The throughput of a closed queueing network with finite node capacities is less than or equal to the throughput of the same network with infinite node capacities, that is,* $\lambda(K) \leq \beta(K)$; $K = 1, \ldots, M$.

**Conjecture 2**

*Probability that a node is empty does not increase as the number of customers in the network increases; that is,* $p_i^K(0) \geq P_i^{K+1}(0)$, $K = 1, \ldots, M - 1$, *where* $p_i^J(0)$ *is the probability that node* $i$ *is empty when there are* $J$ *customers in the network.*

**Conjecture 3**

*Probability that a node is blocked does not decrease as the number of customers in the network increases; that is,* $p_i^{K+1}(b) \geq p_i^K(b)$, $K = 1, \ldots, M - 1$, *where* $p_i^J(b)$ *is the probability that node* $i$ *is blocked when there are* $J$ *customers in the network.*

**Lemma 9**

*Consider a closed exponential queueing network under BAS blocking with parameters given in Table* 1, *and let* $M = \sum_{i=1}^{N} B_i$, $n = \min\{B_i, i = 1, \ldots, N\}$, *and* $\lambda^* = \{\lambda(K), K = 1, \ldots, M\}$. *For a moment, assume that the network has infinite queue capacities and let* $\beta(K)$ *be its throughput when there are* $K$ *customers in it. Then*

$$\beta(n + 1) \leq \lambda^* \leq \beta(M - \min\{B_i, i = 1, \ldots, N\} + 1).$$

*Now, let* $K^*$ *be such that* $\lambda^* = \lambda(K^*)$. *Then*

$$\max(\min\{B_j \text{ such that } p_{ij} \neq 0\ j = 1, \ldots, N\}\ i = 1, \ldots, N) \leq K^* \leq M - \min\{B_i, i = 1, \ldots, N\} + 1.$$

For presentation purposes, consider the closed queueing network illustrated in Figure 10 with parameters given in Table 1, and $B_1 = 3$, $B_2 = 4$, $B_3 = 6$, $B_4 = 5$. Furthermore, assume that the service time at each node is exponentially distributed with rate $\mu_i$. The number of customers in this network can vary from 1 to 18. For $1 \leq K \leq 3$ the network is nonblocking and has a product form steady-state queue length distribution [Gordon and Newell 1967b]. Furthermore, the network has a product form solution with $K = 4$ (Lemma 8) and

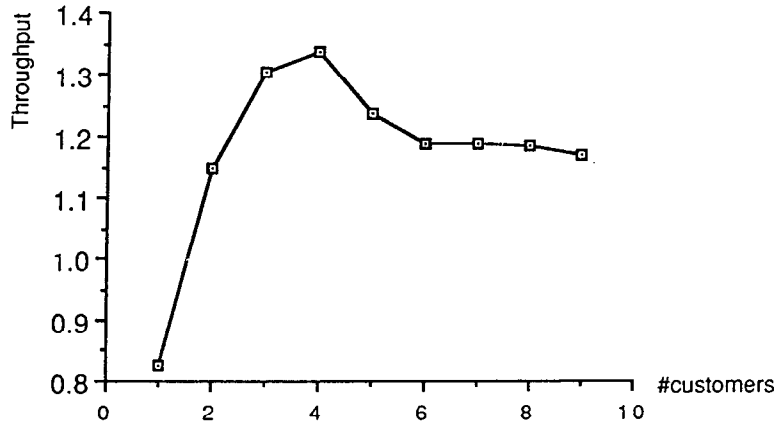


**Figure 19.** Throughout versus number of customers.

$\lambda(K) = \beta(K), K = 1, \ldots, 4$ [Onvural 1987]. Furthermore, $\lambda(K)$ is nondecreasing as $K$ increases from 1 to 4. Hence, the maximum throughput of the network is greater than or equal to the throughput of the network with infinite node capacities and four customers in it, that is, $\lambda^* \geq \beta(4)$. Let $K'$ be the number of customers in the network such that no node can be empty, that is, $K' \geq M - \min\{B_i, i = 1, \ldots, N\} + 1$. For the above example we have $K' \geq 16$. For $16 \leq K' \leq 18$, the probability of any node being empty is equal to zero. Furthermore, $p_i^{K'}(b)$ is nondecreasing in this interval (Conjecture 3). Hence $\lambda_i(K') = \mu_i\{1 - p_i^{K'}(0) - p_i^{K'}(b)\}$ is nonincreasing as $K'$ increases from 16 to 18; that is, $\lambda(16) \leq \lambda(K' + 1)$, $K' = 16, 17, 18$, $\beta(16) \geq (16)$ (Conjecture 1) and $\beta(16) \geq \beta(K)$, $K = 1, \ldots, 15$. Hence, we have $\lambda^* \leq \beta(16)$. Furthermore, if $K^*$ is the number of customers that the throughput of the network is maximum, then $4 \leq K^* < 16$.

We note that these bounds are not usually tight. Nevertheless, these are the only bounds reported in the literature for the maximum throughput of closed networks under BAS blocking. Onvural and Perros [1988] used this result together with the equivalences of open and closed queueing networks with finite capacities to obtain bounds on the throughput of open networks with finite queues. The authors stated in the cited paper that these simple bounds are comparable with other algorithms reported in the literature developed to obtain bounds on the throughput of such networks.

Akyildiz [1988a, b] developed approximation algorithms for the throughput of closed queueing networks with exponential and general service times. He approximates the throughput assuming that the throughput of a blocking network is approximately the same as an equivalent nonblocking network with product form queue length distribution. The equivalent network with infinite queue capacities has the same parameters as the blocking network except $K$. The number of customers in the nonblocking network is chosen such that the number of states of the blocking network is as close to the number of states of the nonblocking network as possible. The only assumption in the algorithm is that the network under consideration should be deadlock free. Akyildiz's algorithm to calculate the throughput of exponential closed queueing networks under BAS blocking can be summarized as follows.

*Algorithm 1*

**S0:** For a given deadlock free closed network under BAS blocking with parameters given in Table 1, calculate the number of states, $Z'$, of the blocking network as follows:

$$Z' = Z_1 * Z_2 * \cdots * Z_N,$$

where $*$ is a convolution operator, and $Z_i$, $i = 1, \ldots, N$, is a $K + 1$ dimensional vector given by

$$Z_i = (z_0^i, z_1^i, \ldots, z_K^i)$$

where

$$z_j^i = \begin{cases} 1 & \text{for } 0 \le j \le B_i + 1 \\ 0 & \text{otherwise} \end{cases}$$

**S1:** Find $q$ such that

$$\left\{ \begin{bmatrix} N - q - 1 \\ N - 1 \end{bmatrix} - Z'(z'_K) \right\}$$

is minimum, where $Z'(z'_K)$ is the $K$th element of the vector $Z'$. Then, $\lambda(K) \approx \beta(q)$, where $\beta(q)$ is the throughput of the network with $q$ customers obtained by solving the network with infinite queues.

We note that

$$\begin{bmatrix} N - q - 1 \\ N - 1 \end{bmatrix} = \frac{(N - q - 1)!}{(N - 1)!\,q!}$$

is the number of states in a closed queueing network with infinite queues and $q$ customers.

The algorithm simply finds a product form network that has approximately the same number of states as the blocking network, and it can be easily implemented. Consider the cyclic network shown in Figure 14 with $N = 3$ and let $B_i = 2$, $i = 1, 2, 3$, and $K = 4$. Then, the state space of this network has 12 states; that is, $Z'(4) = 12$. Now, consider the same network with infinite node capacities. In this case, the state space of the network has 10 states with three customers and 15 states with four customers. Hence, Algorithm 1 produces the throughput of the blocking network to be equal to the throughput of the network with infinite capacities and three customers in it; that is, $\lambda(4) \approx \beta(3)$.

Akyildiz [1988b] applied the algorithm to more than 200 closed networks with different configurations and with different parameters. He reported that in 145 examples, the relative error percentages, RE%, (100[exact throughput − approximate throughput)/exact throughput) was less than 1%. Dallery and Frein [1989] observed that the RE% of this algorithm can go up to 25%. Due to the fact that this algorithm takes into account only the total number of states, the throughput estimates are insensitive to the location of nodes and the service rates. For example, the algorithm would produce the same throughput for two networks with node capacities (1, 5, 1, 5, 1) and (5, 1, 1, 1, 5) while keeping all other parameters the same, although these two networks may have quite different throughputs.

Akyildiz [1988a] applied the same algorithm to closed queueing networks with general service times. In addition to the assumptions of Table 1, assume that the service time at each node is Coxian with two stages; that is, $L = 2$. We note that Algorithm 1 finds a nonblocking network that has the closest number of states as the blocking network under consideration, and this step is independent of the service time distributions at nodes. Once the approximately equivalent nonblocking network is found, the network is solved to obtain its throughput. In case of exponential service time distributions, the nonblocking network has a product form solution and its throughput can be calculated efficiently. When the service times are Coxian, the network does not have a product form solution. Marie [1979] proposed an approximation algorithm to obtain the marginal queue length distributions of a closed queueing network with Coxian service time distributions. The nonblocking network obtained from Algorithm 1 is solved approximately using Marie's method to obtain its throughput. The steps of the algorithm to approximate the throughput of a closed queueing network under BAS blocking with Coxian service time distributions are given below.

*Algorithm 2*

**S0:** Given a deadlock free closed queueing network find a nonblocking network with $K'$ customers using Algorithm 1.

**S1:** Solve the nonblocking network with Coxian service times approximately using Marie's method and calculate the throughput $\beta(q)$ of the network. Then, the throughput of the blocking network is

assumed to be equal to $\beta(K)$. That is, $\lambda(K) \approx \beta(q)$.

The accuracy of this algorithm is not as good as that of Algorithm 1. This is due to the fact that Marie's method introduces an additional error to the error produced by Algorithm 1. Still, the RE% reported in the paper was less than 10%. Finally, we note that these two approximations can be used only to calculate the throughput. In particular, both Algorithms 1 and 2 do not produce accurate results for other performance measures such as mean queue lengths and marginal queue length probabilities [Onvural 1987].

Another approximation algorithm for the throughput of closed queueing networks under BAS blocking was developed by Suri and Diehl [1984]. Their algorithm is also applicable to networks under BBS-SO blocking, and it will be presented in Section 3.2.

Onvural and Perros [1989b] developed an approximation algorithm to calculate the throughput of large closed exponential queueing networks with finite queues. The main steps of the algorithm are given below.

*Algorithm 3*

**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 obtain its throughput $\lambda(K^*)$.

**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 the above calculated points.

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

The critical step in the algorithm is finding the number of customers, $K^*$, that produces the maximum throughput $\lambda(K^*)$. In closed queueing networks with exactly one node with an infinite capacity, $K^*$ can be found exactly using the following result [Onvural and Perros 1989a].

**Lemma 10**

*Consider an exponential closed queueing network under BAS blocking. Let m be the index of the node with the maximum capacity; that is,* $B_m = \max(B_i, i = 1, \ldots, N)$, *and* $M = \sum_{i=1}^{N} B_i$ *be the total capacity of the network. If* $B_m \geq M - B_m$, *then the network has the same throughput for all* $K \in S$, *where* $S = \{L: M - B_m + 1 \leq L \leq B_m + 1\}$.

For presentation purposes, let node 1 be the node with an infinite capacity. Then, $B_1 > M - B_1$. Let $M'$ denote the capacity of nodes two to $N$ (the sum of node capacities of all nodes other than the node with the infinite capacity) plus 1; that is, $M' = \sum_{i=2}^{N} B_i + 1$. From Lemma 10, the throughput of the network is the same for all $K \geq M'$. Using the monotonicity of the curve with respect to the number of customers [Persone and Grillo 1987; Onvural and Perros 1989b], we have $K^* = M' + 1$. Algorithm 3 was also applied to cyclic networks with finite capacities. In this case, it was assumed that the maximum throughput occurs at $K^* = (M + N)/2$, which is approximated from the same network under BBS-SO blocking. The interested reader may refer to Onvural and Perros [1989b] for details. Another drawback of the algorithm is solving the network numerically with $K^*$ customers to calculate $\lambda(K^*)$, since this is a time-consuming task. Still, the algorithm results in savings of 50 to 85% of CPU time as compared to solving the network numerically, and it produces fairly accurate results. The savings in CPU time increases with the size of the network. Of the 100 examples run by the authors, the relative error percentages was observed to be less than 5%.

The throughput of cyclic networks with $M = \sum_{i=1}^{N} B_i$ customers can be calculated efficiently using the following lemma [Onvural and Perros 1989b].

**Lemma 11**

*Consider an exponential cyclic network under BAS blocking with node capacities* $B_i$ *and K customers. If* $K = M$, *then the throughput of the network is equal to* $1/\mathrm{E}[\max(X_1, X_2, \ldots, X_N)]$, *where* $X_i$ *is the*

*service time at node i. Furthermore, assuming $X_i$'s are distributed exponentially with rate $\mu_i$, we have*

$$\mathrm{E}[\max(X_1, X_2, \ldots, X_N)] = \int_0^\infty \left(1 - \prod_{i=1}^{N} (1 - e^{\Sigma \mu_i t})\right) dt.$$

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 $X_1 \leq X_2 \leq X_3$. Furthermore, assume that at $t = 0$ all servers are busy working. Then at $t = X_3$, all three servers will become blocked and a deadlock will occur. If we assume that deadlocks are detected immediately and resolved by instantaneously exchanging the blocked customers, then at $t = X_3$ customer at note 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 at which all three servers start a new service are the renewal points and the throughput of the cyclic network is 1/(expected time between arrivals) by definition.

### 3.1.2 Mean Queue Lengths

To the best of our knowledge, there are only two approximations reported in the literature to calculate the mean queue lengths of arbitrary closed queueing networks under BAS blocking: Akyildiz [1988c, 1989a]. In particular, Akyildiz [1988c] developed an approximation to calculate the mean queue lengths and the throughput of each node. The algorithm, however, needs more work since it does not produce accurate errors. The other algorithm [Akyildiz 1989a] is not time efficient but produces fairly accurate results. The approximation algorithm first obtains the steady-state joint queue length probabilities of the network as if there is no blocking (i.e., with infinite node capacities). The steady-state joint queue length probabilities of the blocking network are then approximated by using a transformation from the states of the nonblocking network to the states of the blocking network. We note that this algorithm should be used only to calculate the mean queue lengths since it does not produce accurate results for the throughput or the marginal queue length probabilities. Let $(k_1, k_2, \ldots, k_N)$ be the state of the network under consideration with infinite buffer capacities, where $k_j$ is the number of customers at node $j$. Then, the network has a product form queue length distribution if the service times are exponentially distributed [Gordon and Newell 1967b]. In the case of general servers, the algorithm is applicable by first applying Marie's method [1979], similar to Algorithm 2.

Transforming the states of the network with infinite capacities to the states of the blocking is done as follows: For any state $(k_1, k_2, \ldots, k_N)$ of the network with infinite capacities, if there exists a node $i$ with $k_i > B_i$, then

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

Hence, if the capacity of node $i$ 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) (referred to as the normalization step). $e_i$ is the relative number of visits a customer makes to the $i$th node and is given by Equation (7). This transformation is applied until the number of customers at each node is less than or equal to the respective node capacities, giving the normalized state. Consider the cyclic network shown in Figure 14 with $N = 3$, $B_1 = 2$, $B_2 = 1$, $B_3 = 2$, and $K = 4$. Furthermore, assume that the service time at each node is exponentially distributed. We note that $e_i = 1$, $i = 1, 2, 3$. For a moment, consider the network with infinite node capacities and the state (0, 0, 4), where nodes 1 and 2 are empty and there are four customers at node 3. Since the capacity of node 3 is exceeded, the first normalization gives the state (0, 2, 2). As the capacity of node 2 is exceeded, the normalization step is applied once more giving the state

(1, 1, 2) in which no node capacity is exceeded. Hence, the state (0, 0, 4) of the nonblocking network becomes (1, 1, 2) after the normalization step is applied twice.

Let $f(\underline{k})$ denote the normalized state where the $i$th component of $f$, that is, $f_i(\underline{k})$ is the number of customers at node $i$ after the normalization step. Then the mean queue length of node $i$, $L_i$, is equal to $\sum_{k\text{feas}} f_i(k)p(\underline{k})$, where $p(\underline{k})$ is the steady-state joint queue length distribution of the network with infinite queue capacities. The algorithm was applied to a variety of closed queueing networks under BAS blocking. It was reported in Akyildiz [1989a] that the maximum relative error percentage observed was 20%.

Dallery and Frein [1989] presented an approximation technique for the analysis of cyclic networks in which there is at least one node with an infinite capacity. The approach is similar to the ones developed in the literature for open queueing networks with blocking. The algorithm decomposes the network into individual nodes with revised capacities, revised service rates, and revised arrival rates. In particular, if node $i$ causes the blocking of the preceding node, that is, $B_i < K$, then its capacity is increased by 1 to accommodate the blocked unit. If $B_i \geq K$, then its capacity is set to be equal to $K$. The service process is revised to reflect the possible delay a customer might undergo due to blocking. Upon completion of its service, a customer at node $i$ may find node $i + 1$ full. Then the blocked customer will wait until a departure occurs from node $i + 1$. We note that the blocking delay is not necessarily equal to the remaining service time at node $i + 1$ as node $i + 1$ may get blocked by node $i + 2$, node $i + 2$ may get blocked by node $i + 3$, and so on. Hence, the mean delay, $1/\mu_i^*(n)$, at node $i$, when there are $n$ customers in it is equal to

$$\frac{1}{\mu_i^*}(n) = \frac{1}{\mu_i} + b_i^*(n)t_i^*(n) \qquad (19)$$

where $t_i^*(n)$ is the mean blocking delay, $b_i^*(n)$ is the probability that a customer upon service completion will find the successor node full and node $i$ is blocked. It was assumed that


**Figure 20.** Node $i$ in isolation.

$$b_i^*(n) = \begin{cases} \dfrac{P_{i+1}(n)}{1 - P_{i+1}(B_{i+1} + 1)} \\ \quad \text{if } K - n - B_{i+1} \geq 0 \\ 0 \quad \textit{otherwise} \end{cases} \qquad (20)$$

where $P_{i+1}(n)$ is the marginal probability of having $n$ customers at node $i + 1$. Equation (20) is the probability that an arriving customer finds $n$ customers in an M/M/1/$B_{i+1}$+1/FCFS queue [Cohen 1969]. It was used in the algorithm to approximate the blocking probability. It is an approximation because neither the arrival distribution nor the time between departures from node $i$ is exponential in the original system (due to blocking). The blocking delay was assumed to be equal to the mean service time of the successor queue, that is, $t_i^*(n) = 1/\mu_{i+1}$. That is, if node $i$ is blocked by node $i + 1$ then it was assumed that node $i + 1$ cannot get blocked by node $i + 2$, even if this may not be the case. Then a node in isolation looks like an M/M/1/$B_i$+1/FCFS queue in Figure 20.

Clearly, the overall arrival rate $\beta_i$ is not known. If the throughput of node $i$, $\lambda_i(K)$, is known, however, then $\beta_i$ can be calculated as a fixed point problem using the following algorithm.

*Algorithm 4*

**S0:** Let $\beta_i = \lambda_i(K)$.

**S1:** Solve the M/M/1/$B_i$+1/FCFS queue to obtain $P_i(B_i + 1)$, the marginal probability of having $B_i + 1$ customers at node $i$.

**S2:** Let $\Omega = \lambda_i(K)/(1 - P_i(B_i + 1))$. If $|\Omega - \beta_i| < \epsilon$ then STOP; else set $\beta_i = \Omega$ and go to **S1**.

This procedure has been used in numerous algorithms in the context of open

queueing networks with blocking [Altiok and Perros 1987].

We are now ready to present Dallery and Frein's [1989] algorithm that calculates the throughput of the network and the mean queue length of each node.

*Algorithm 5*

**S0:** Set the bounding values for the throughput, $\lambda(K)$, of the network. Note that in cyclic networks, throughput of nodes and the throughput of the network are the same because $e_i = 1$, $i = 1, \ldots, N$. Let $\lambda_{\min} = 0$ and $\lambda_{\max} = \min\{\mu_i, i = 1, \ldots, N\}$.

**S1:** Let $\lambda(K) = (\lambda_{\min} + \lambda_{\max})/2$.

**S2:** For $i = N$ down to 1
(* for each node in the network do *)
calculate $b_i^*(n)$ using Equation (9)

set $\mu_i^*(n) = (\mu_i^{-1} + b_i^*(n)/\mu_{i+1})^{-1}$

solve this $M/M/1/B_i+1$ queue using Algorithm 5
(Note: If $B_i \geq K$, then the node capacity is set to be equal to $K$).

**S3:** Calculate the average length, $L_i$, of node $i$, $i = 1, \ldots, N$ to obtain $L = \sum_{i=1}^{N} L_i$.

**S4:** If $|L - K| < \epsilon$ then STOP;
else if $L > K$ then set $\lambda_{\max} = \lambda(K)$ else if $L < K$ then set $\lambda_{\min} = \lambda(K)$
go to **S1**.

This algorithm was proven in the paper to be convergent. It produces both the throughput of the network and the mean queue lengths. Although the algorithm calculates these performance measures from the marginal queue length probabilities, their accuracy was not reported. The average relative error percentages for the other two performance measures were reported to be 4.2%, whereas the maximum RE% was observed to be equal to 20%.

### 3.1.3 *Blocking Network as an Approximate BCMP Node*

Perros et al. [1988] developed a numerical procedure for the approximate analysis of closed queueing networks in which some of the queues have finite capacities. The approximation algorithm is based on Norton's theorem [Chandy et al. 1975].

*Algorithm 6*

**S0:** Group all the finite queues and those infinite queues that are liable to getting blocked into a subnetwork (blocking subnetwork) and the remaining ones into another subnetwork (nonblocking subnetwork).

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

**S2:** Construct a composite queue with a state dependent throughput equal to $T(n)$, $n = 1, 2, \ldots, K$. Now, in the original network substitute the nonblocking 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 $p_c(n)$, $n = 1, \ldots, K$ be the marginal queue length probability that there are $n$ units in the composite queue as calculated in S2. Then, from S1, we have $p(\underline{m}) = p(\underline{m} \mid n)p_c(n)$, $\underline{m} \in S_n$ and $n = 1, 2, \ldots, K$. Using $p(\underline{m})$, the marginal queue length distribution for each queue in the nonblocking subnetwork can be easily obtained.

Numerical investigation in the paper shows that the algorithm is very accurate. In particular, the authors observed that the throughput and the mean queue legnths have relative errors less than 1%. Also, the relative error percentages on the queue length probabilities were observed to be less than 5%. The drawback of the algorithm is solving the blocking network numerically, which is time consuming for large networks.

### 3.2 Blocked before Service Blocking

The BBS blocking mechanism was introduced to model computer and telecommunications systems. In this blocking

mechanism, the lack of a space in the destination node forces the server to suspend its service. Our discussion is limited to deadlock free networks, since closed queueing networks under BBS blocking in which deadlocks can occur have not been studied in the literature.

### 3.2.1 BBS-SNO Blocking

This blocking mechanism was introduced in open networks and, to the best of our knowledge, no study of closed queueing networks with this form of blocking has been reported in the literature. However, since BBS-SNO blocking is equivalent to BAS blocking in cyclic exponential networks (Lemma 3b), results reported for cyclic networks under BAS blocking are readily applicable to cyclic networks under BBS-SNO blocking after the node capacities are adjusted.

### 3.2.2 BBS-SO Blocking

Closed queueing networks under BBS-SO blocking were first studied by Gordon and Newell [1967a] in the context of cyclic networks. The service time at each node is assumed to be exponentially distributed. First, we will discuss the concept of *holes* as introduced by Gordon and Newell. 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 total number of holes in the network is equal to $\sum_{j=1}^{N} B_j - K$. As the customers move sequentially through the cyclic network, the holes execute a counter sequential motion since each movement of a customer from the $j$th node to the $(j + 1)$st node corresponds to the movement of a hole in the opposite direction (i.e., from the $j + 1$st node to the $j$th node). It is then shown that these two networks are *duals*. That is, if a customer (hole) at node $j$ is blocked in one system, then node $j + 1$ has no holes (customers) in its dual. Let $(B_i, \mu_i)$ be the capacity and the service rate of node $i$ and $\{(B_1, \mu_1), (B_2, \mu_2), \ldots, (B_N, \mu_N)\}$ be a cyclic network with $K$ customers. Then its dual is $\{(B_1, \mu_N), (B_N, \mu_{N-1}), \ldots, (B_2, \mu_1)\}$ with $\sum_{j=1}^{N} B_j - K$ customers. Let, $p(\underline{n})$ and $p^D(\underline{n})$ be the steady-state queue length probabilities of a cyclic network and its dual, respectively, where $\underline{n} = (i_1, i_2, \ldots, i_N)$ is the state of the network with $i_j$ being the number of customers at node $j$. Then for all feasible states, we have

$$p(i_1, i_2, \ldots, i_N) = p^D(B_1 - i_1\ B_N - i_N, \ldots, B_2 - i_2).$$

We note that if the number of customers in the network is such that no node can be empty, then the dual network is a nonblocking network (i.e., the number of holes is less than or equal to the minimum node capacity) and it has a product form queue length distribution [Gordon and Newell 1967b]. But then, from the concept of duality, the original network has a product form queue length distribution. Hence, we have the following lemma [Gordon and Newell 1967a]:

**Lemma 12**

*Consider a cyclic network under BBS-SO blocking. The service time at each node is assumed to be exponentially distributed. The network has a product form queue length distribution if*

$$K \geq \sum_{i=1}^{N} B_i = \min\{B_j, j = 1, \ldots, N\}.$$

Furthermore, the following conjecture is a consequence of duality in cyclic networks [Onvural 1987; Persone and Grillo 1987].

**Conjecture 4**

*An exponential network under BBS-SO blocking has the same throughput with $K$ and $\sum_{j=1}^{N} B_j - K$ customers in it.*

This conjecture was proved in the case of symmetrical node capacities, that is, $B_i = B, i = 1, \ldots, N$, and its validity for arbitrary node capacities was observed empirically. In addition to the above conjecture, it was observed that the throughput of cyclic networks under BBS-SNO blocking is nondecreasing as $K$ increases from 1 to $K' = [\sum_{i=1}^{N} B_i/2]$ and nonincreasing as $K$ increases from $K'$ to $\sum_{i=1}^{N} B_i - 1$ [Onvural

and Perros 1989b; Persone and Grillo 1987].

Shanthikumar and Yao [1989] considered exponential cyclic queueing networks under BBS-SO blocking and identified the conditions under which the performance measures are monotone in service rate, node capacity, and population size. Let $\mu_i(k)$ be the load dependent service rate at station $i$. Furthermore, let $\underline{B} = (B_1, \ldots, B_N)$ and $\underline{\mu} = (\mu_1(k), \ldots, \mu_N(k))$ be the vector of node capacities and service rates, respectively. The main properties obtained by Shanthikumar and Yao are given as follows.

**Lemma 13**

*Consider a cyclic network under BBS-SO blocking with two sets of service rates,* $\mu_i^1(k)$, $\mu_i^2(n)$. *If* $\mu_i^1(k) \geq \mu_i^2(n)$, $k \geq n$, $i = 1, \ldots, N$, *then*

$$\text{Throughput}(\underline{\mu}^1, \underline{B}, \underline{K}) \geq \text{Throughput}(\underline{\mu}^2, \underline{B}, \underline{K}).$$

**Lemma 14**

*Consider a cyclic network under BBS-SO blocking with two sets of buffer capacities,* $\underline{B}^1$, $\underline{B}^2$ *and assume that* $\mu_i(k)$ *is increasing in k for each i,* $i = 1, \ldots, N$. *If* $\underline{B}^2 \geq \underline{B}^2$ *then*

$$\text{Throughput}(\underline{\mu}, \underline{B}^1, K) \geq \text{Throughput}(\underline{\mu}, \underline{B}^2, K).$$

**Lemma 15**

*Let* $B^* = \max\{B_i; i = 1, \ldots, N\}$. *Then, for* $0 < K < B^*$,

$$\text{Throughput}(\underline{\mu}, \underline{B}, K + 1) \geq \text{Throughput}(\underline{\mu}, \underline{B}, K).$$

Lemma 15 states that throughput is nondecreasing with respect to the number of customers as long as the number of customers is less than the maximum node capacity in the network. Similarly, Lemmas 13 and 14 show the monotonicity of the throughput with respect to service rates and node capacities, respectively. In particular, Lemma 13 states that the throughput of the network does not decrease if the service rate of a node (or a group of nodes) increases. Similarly, Lemma 14 states that the throughput of the network is nondecreasing as the buffer capacity of a node (or a group of nodes) increases.

Approximation algorithms for cyclic networks under BBS-SO blocking were proposed by Suri and Diehl [1986] and Onvural and Perros [1989b]. In particular, Onvural and Perros used the above conjecture and assumed that the maximum throughput (w.r.t. $K$) is achieved with $K^* = [\sum_{i=1}^{N} B_i/2]$ customers. With this value of $K^*$, Algorithm 3 is readily applicable to cyclic networks under BBS-SO blocking. The algorithm was also applied to the central server model with exactly one node with an infinite capacity, using a result similar to Lemma 8 [Onvural and Perros 1989a]. The algorithm, in general, was observed to work better in BBS-SO blocking than in BAS blocking. This is because the above value of $K^*$ was observed empirically to be exact for closed queueing networks under BBS-SO blocking (Conjecture 4).

Suri and Diehl [1986] introduced the concept of *variable buffer size* and used it together with the *flow equivalent approximations* to approximate the throughput of cyclic networks with at least one node with an infinite capacity. The service time at each node is assumed to be exponentially distributed with rate $\mu_i$, $i = 1, \ldots, N$. Without loss of generality, let the node with the infinite capacity be node 1. In the flow equivalent approach [Chandy and Sauer 1978; Chandy et al. 1975] all nodes other than node $i$, for some $i$, are replaced by a single composite server with state dependent service rates $\mu_i(j)$, where $j$ is the number of customers in the composite queue. When this approach is used in networks with finite queues, the capacity of the composite node plays an important role. If we use a node capacity of $B = \sum_{j=i+1}^{N} B_j$ (i.e., the total capacity of the downstream nodes), 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 nodes $i + 1$ to $N$, not all of them need to be at node $i + 1$. Thus, server $i$, $S_i$, *sees* a finite capacity of size $k$,

$B_{i+1} \leq K \leq \sum_{j=i+1}^{N} B_j$, in the composite node for some fraction of time. The variable buffer size model introduced by Suri and Diehl is an attempt to capture this view.

Let $p(k \mid K)$ be the fraction of time the composite queue behaves like a $k$-buffer node (including the server). Given the fixed node capacity $k$ and state dependent service rates, the two node network has a product form queue length distribution; hence it can be solved efficiently. If $p(k \mid K)$'s are known for all $k$, then the performance measures of the original network can be calculated as a weighted sum of the performance measures of the two node networks. The approximation algorithm is based on the idea that the $i$th server can view all the downstream nodes $i + 1$ to $N$ in terms of a single finite node that is the flow equivalent representation of the downstream servers. In particular, node 2 of the variable buffer size model (Figure 21) is this flow equivalent node. The service rate at node 1 of the variable buffer size model is that of node $i$; that is, $\mu_i$. The total number of customers is varied from 1 to $K$ for each node $i$. The weights $p_i(k \mid K)$ are obtained by calculating the probabilities that there are $b_i$ customers at node 1 and $k - b_i$ customers at node 2. The algorithm proceeds in this fashion moving upstream until all nodes have been considered. It is easily started since the $(N - 1)$st node sees the $N$th node as the flow equivalent node and as the destination of the $N$th node (i.e., node 1 has an infinite capacity).

The complete algorithm is given in Suri and Diehl [1986]. Validation tests presented in the cited paper show that the algorithm is accurate and fast. These tests, however, are restricted to three-node cyclic networks. Of the 100 examples reported in Diehl [1984], 57% had RE% less than 1% with the maximum RE% being equal to 7%. The accuracy needs to be investigated for networks with more than three nodes.

A similar algorithm is given for cyclic networks under BAS blocking. It was discussed in Diehl [1984] that the algorithm can be used to approximate the other performance measures; it is also applicable to arbitrary topologies as well as open networks. No validation tests have been reported for these cases.

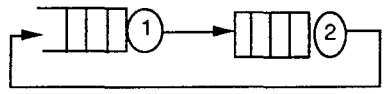


**Figure 21.** Two-node variable buffer size model.

Finally, we note that, to the best of our knowledge, there is no algorithm reported in the literature to calculate the mean queue lengths of closed queueing networks under BBS-SO blocking.

### 3.3 Repetitive Service Blocking

The RS blocking was introduced by Caseau and Pujolle [1979] in open tandem networks and by Pittel [1979] 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 node was full. Recently, it has been used in modeling flexible manufacturing systems [Yao and Buzacott 1985a, b, c; 1986].

#### *3.3.1 RS-FD Blocking*

We first note that RS-FD blocking is equivalent to BBS-SO blocking independent of the topology of the network if the service time at each node is exponentially distributed (Lemma 2). Thus, exact and approximate results presented for BBS-SO blocking in Section 3.2.2 in the context of cyclic networks 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 RS-FD is equivalent to RS-RD, by definition. Hence, results discussed next in Section 3.3.2 are applicable to this blocking mechanism. Other than these equivalences, there are no results reported in the literature on closed queueing networks under RS-FD blocking.

#### *3.3.2 RS-RD Blocking*

Let us consider a closed queueing network under RS-RD blocking with parameters given in Table 1. The service time at node $i$ is exponentially distributed with mean

$1/\mu_i$. Let $f_i(n_i)$ bed the rate at which the server at the $i$th node works when there are $n_i$ customers in the node. 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. We note that in RS-RD blocking, $b_j(B_j) = 0$ and $b_j(n_j) = 1$ for $0 \le n_j < B_j$. Let $\pi(i_1, i_2, \ldots, i_N)$ be the steady-state queue length distribution of a closed network, where $i_j$ is the number of customers at node $j$, $\sum_{j=1}^{N} i_j = K$ and $i_j \le B_j, j = 1, \ldots, N$. In networks with reversible routing, it is shown that [Balsamo and Iazeolla 1983; Dallery and Yao 1986; Hordijk and Van Dijk 1981; Pittel 1979; Akyildiz and Von Brand 1989a, b, c]:

$$\pi(i_1, i_2, \ldots, N) = C \prod_{j=1}^{N} \lambda_j \prod_{k=1}^{i_j} \frac{b_j(k-1)}{\mu_j(k) f_j(k)},$$

where $C$ is the normalizing constant to ensure that the sum of the steady-state queue length probabilities is equal to 1.

Lemma 16, proved in Onvural [1989b], unifies the product form queue length distributions reported in the literature for closed queueing networks under RS-RD blocking.

**Lemma 16**

*Any closed queueing network under RS-RD blocking is a truncated process (in which no more than $B_i$ jobs are allowed at node $i$) of the same network with infinite buffer capacities. Hence, a closed queueing network with a reversible routing under RS-RD blocking has a product form queue length distribution if the network with infinite buffer capacities has a product form queue length distribution.*

When the routing matrix is not reversible, the closed queueing networks under RS-RD blocking have been shown to have product form queue length distributions in the following two cases (Hordijk and Van Dijk 1982]:

(a) The probability, $b_j(n_j)$, that a customer will be admitted to a node $j$ is constant, independent of $n_j$; that is, $b_j(n_j) = b_j$, $j = 1, \ldots, N$. In this case, blocking of a node occurs independent of the number of customers in the destination node. We note that this definition of blocking does not correspond to any of the blocking mechanisms defined in Section 1, since the blocking of a node in those blocking mechanisms is caused only if node $j$ if full, that is, $b_j(B_j) = 1$ and $b_j(n_j) = 0$, $n_j = 0, \ldots, B_j - 1$.

(b) The rate $f_i(n_i)$ at which server $i$ works is constant, independent of $n_i$; that is, $f_i(n_i) = f_i$. We note that, in cyclic networks, this result is immediate from Gordon and Newell's [1967a] result of duality in BBS-SO blocking and from the equivalences of the two blocking mechanisms, since a constant rate, $f_i$, means that no node in the network can be empty, that is, $K > \sum_{i=1}^{N} B_i$. Akyildiz and Von Brand [1987a] extended this result to include $K = \sum_{i=1}^{N} B_i$.

Consider a closed queueing network with parameters given in Table 1 with a reversible routing. The service time at each node is assumed to be exponentially distributed. The normalization constant and the performance measures of such networks can be calculated efficiently using Algorithm 8 [Akyildiz 1989b]. Define an auxiliary function $g(k, n)$, where $k$ denotes the number of jobs and $n$ denotes the number of stations. Then the normalizing constant, $g(K, N)$, can be calculated as follows:

*Algorithm 7*

**S0:** Let $g(0, n) = 1$, $n = 1, \ldots, N$. Compute:

$$g(k, 1) = \begin{cases} x_1^k & \text{if } k \le B_1 \\ 0 & \text{if } k > B_1 \end{cases}$$

**S1:** $g(k, n) = g(k, n-1) + x_N g(k-1, n) - x_n^k h(m, n)$, where

$$h(m, n) = \left(\frac{1}{m}\right) \sum_{i=1}^{N} \sum_{j=1}^{m} x_i^j \, h(m-j, n),$$

with $h(0, n) = 1$, $n = 1, \ldots, N$ and $x_i = e_i/\mu_i$, where $e_i$ is the visit ratio and $\mu_i$ is the service rate of node $i$, respectively.

The function $h(m, n)$ eliminates the nonfeasible states (i.e., $k > B_n$). Once the normalization constants are obtained, the performance measures of reversible networks under RS-RD blocking are

calculated as follows:

(i) The steady-state queue length distribution:

$$p(n_1, \ldots, n_N) = \left( \prod_{i=1}^{N} x_i^{n_i} \right) \Big/ g(K, N),$$

where $n_i$ is the number of customers at node $i$, $0 \le n_i \le B_i$.

(ii) Marginal queue length distribution of node $i$:

$$P_i(n) = \frac{x_i^n G_i(K-n)}{g(K, N)}, \quad i = 1, \ldots, N,$$

where $G_i(K-n)$ is the normalization constant computed without considering the $i$th station with $K-n$ customers in the network.

(iii) Mean queue length of node $i$:

$$L_i = \sum_{n=1}^{B_i} n \frac{x_i^n G_i(K-n)}{g(K, N)}, \quad i = 1, \ldots, N.$$

(iv) Throughput of node $i$:

$$\lambda_i = \frac{e_i}{g(K, N)} \left( \sum_{j=1 \& j \neq i}^{N} p_{ij} H_{ij}(K-1) \right), \quad i = 1, \ldots, N,$$

where $H_{ij}(K) = b_i(K) * b_j(K) * \prod_{t \neq i;j} a_t(K)$ with initial values $H_{ij}(0) = 1$; "$*$" is the convolution operator, and $a_i(K) = (1, x_i^1, x_i^2, \ldots, x_i^{B_i})$; $b_i(K) = (1, x_i^1, x_i^2, \ldots, x_1^{B_i - 1}, 0)$.

(v) The effective utilization: $\mu_i^E = \lambda_i/\mu_i$; the total utilization: $u_i^T (1 - P_i(0))\mu_i$, $i = 1, \ldots, N$.

(vi) Mean waiting time: $w_i = L_i/\lambda_i i$, $i = 1, \ldots, N$.

Yao and Buzacott [1985a] considered closed queueing networks under RS-RD blocking in which the routing probabilities from node $i$ to node $j$ are state dependent. In particular, let us consider reversible networks studied above assuming $p_{jk}$ depends on $i_j$ and $i_k$ as follows:

$$p_{jk}(i_j, i_k) = \phi_j(i_j)\psi_k(i_k),$$

where $\phi_j(i_j)$ and $\psi_k(i_k)$ are arbitrary functions such that $\phi_j(i_j) > 0$ if $i_j > 0$, $\psi_k(i_k) > 0$ if $i_k > 0$ with $\phi_j(0) = 0$ and $\psi_k(B_k) = 0$. Under these state dependent routing probabilities, a queueing network has a reversible routing and has the following product form steady-state queue length distribution:

$$\pi(i_1, i_2, \ldots, i_N) = C \prod_{j=1}^{N} \psi_j(k-1) \prod_{k=1}^{i_j} \frac{1}{\mu_j(k) f_j(k) \phi_j(k)},$$

where $C$ is the normalizing constant. The effect of this state dependent routing is as follows: Upon completion of its service at node $i$, a customer will probabilistically join any of the destination nodes that at that moment, are not full. If all the destination nodes are full, the service will be repeated at node $i$ (i.e., RS-RD 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 $\sum_m \phi_j(i_j)\psi_m(i_m) = 1$, or equivalently, $\phi_j(i_j) = 1/\sum_m \psi_j(i_m)$. Now, let $\psi_m(i_m) = B_m - i_m$. Then, $\phi_j(i_j) = 1/\sum_m B_m - (K - i_j))$, with $m \neq j$. Thus, we have the following "probabilistic shortest queue" routing

$$p_{jk}(i_j, i_k) = \frac{B_k - i_j}{\sum_m B_m - (K - i_j)}$$

for all $j \neq k$ such that $m \neq j$.

In this type of routing, a customer may join a node that has the shortest queue with the highest priority. A customer never joins a node that is full; hence, no blocking can take place in the network. This is an extension of RS-RD blocking to nonblocking networks with reversible routing.

Yao and Buzacott [1985c] reported an approximation algorithm for analyzing closed queueing networks under RD-RS blocking. In addition to the parameters given in Table 1, assume that each queue $i$ 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 approximated by an exponential distribution with the same mean as the Coxian server, then the resulting exponential network is reversible and has a product form queue length distribution. The approximation is based on the notion of exponentialization. The main steps of the algorithm are as follows.

*Algorithm 8*

**S0:** For each node $i$, substitute an exponential server with the same rate $\mu_i(n_i)$ as the original 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 $i$, $i = 1, \ldots, N$, using

$$\lambda_i(n_i) = \frac{\mu_i(n_i + 1)p_i(n_i + 1)}{p_i(n_i)}, \qquad n_i = 0, \ldots, B_i.$$

**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 in Step 2. Other parameters of the queue are the same as in the original network. For each node $i$, obtain the marginal queue length probabilities $q_i(n_i)$, $n_i = 0, \ldots, B_i$; $i = 1, \ldots, N$.

**S4:** Derive state dependent service rates $v_i(n_i)$ as follows:

$$v_i(n_i) = \frac{\lambda_i(n_i - 1)q_i(n_i - 1)}{p_i(n_i)}, \qquad n_i = 0, \ldots, B_i; \quad i = 1, \ldots, N.$$

**S5:** If $\max|v_i(n_i) - \mu_i(n_i)| < \epsilon$ then STOP. Else set $\mu_i(n_i) = v_i(n_i)$ for all $n_i = 0, \ldots, B_i$ and $i = 1, \ldots, N$ and goto **S1**.

The algorithm produces the marginal queue length probabilities. Validation examples in the paper show that the accuracy of the algorithm is good.

Kouvatsos and Xenios [1989] used the principle of maximum entropy to find an approximate product form queue length distribution for closed queueing networks under RS-RD blocking. The algorithm requires the solution of nonlinear equations using the principle of maximum entropy. An interested reader may refer to Kouvatsos [1983] for a detailed description of the maximum entropy principle. The procedure is based on decomposing the network into individual nodes and analyzing them in isolation. In particular, each node in isolation is studied as a GE/GE/1/B/FCFS queue, that is, with generalized exponential arrival and service distributions.

Let $P_i(n)$ be the marginal probability of having $n$ customers in a GE/GE/1/B/FCFS queue. Then,

$$P_i(n) = P_i(0)g^{h(n)}x^n y^{f(n)}, \quad n = 0, 1, \ldots,$$

where $h(n) = \min(1, \max(0, n))$, $f(n) = \max(0, n - B + 1)$, $g = e^{-\beta_1}$, $x = e^{-\beta_2}$, $y = e^{-\beta_3}$. $\beta_i$, $i = 1, 2, 3$ are obtained by solving the following optimization problem:

$$\text{Max } H(p) = \sum_{n=0}^{B_i} P_i(n) \log P_i(n)$$

subject to

(1) (Normalization) $\sum_{n=0}^{B_i} P_i(n) = 1.$

(2) (Utilization)

$$\sum_{n=1}^{B_i} P_i(n) = \rho = \sum_{n=0}^{B_i} h(n)P_i(n).$$

(3) (Mean number of customers)

$$\sum_{n=0}^{B_i} nP_i(n) = L_i.$$

(4) (The probability that node $i$ is full)

$$\sum_{n=0}^{B_i} f(n)P_i(n) = \phi.$$

Consider a closed queueing network under RS-RD blocking. The service time at each node follows a generalized exponential with mean $1/\mu_i$ and squared coefficient of variation $c_{si}^2$. The service distribution of each queue is revised to accommodate the delays a customer might undergo due to blocking. In particular, in this blocking a customer upon completing its service at node $i$ attempts to join node $j$. If node $j$ at that moment is full, then the customer goes back to $i$th server and receives another service. This is repeated until the customer completes its service at a time that there is a space in its destination. Hence, the effective service time of a customer is a random number of $GE$ service times. Let $\pi_i$ be the probability that the customer will find the destination node full. Then $\pi_i$ is approximated as follows:

$$\pi_i = \sum_{j=1}^{N} p_{ji}P_j(B_j), \tag{21}$$

where $P_j(B_j)$ is the marginal probability of queue $j$ being full. Then the effective service at node $i$ is represented by a $GE$ distribution with

$$\mu_i^* = \mu_i(1 - \pi_i)$$

and

$$c_{si}^{*2} = \pi_i + c_{si}^2(1 - \pi_i). \tag{22}$$

Similarly, the arrival process to node $i$ in isolation is the superposition of the departure processes from nodes $j$ with $p_{ji} > 0$, each with mean, $E(d_i)$, and squared coefficient of variation, $c_{d_i}^2$ as follows:

$$\begin{aligned} E(d_i) &= \frac{1}{\lambda_i} \\ c_{d_i}^2 &= \left(\frac{\lambda_i}{\mu_i^*}\right)\left(1 - \frac{\lambda_i}{\mu_i^*}\right) + \left(\frac{\lambda_i}{\mu_i^*}\right)c_{si}^{*2} \\ &\quad + \left(1 - \left(\frac{\lambda_i}{\mu_i^*}\right)\right)c_{ai}^2, \end{aligned} \tag{23}$$

where

$$c_{ai}^2 = -1 + \sum_{j=1}^{N} \frac{\lambda_j p_{ji}}{\lambda_i}\{c_{dji}^2 + 1\},$$

and

$$c_{dji}^2 = 1 - p_{ji} + p_{ji}c_{dj}^2.$$

In case of open networks, the algorithm is easily started with some initial values, and then the procedure iterates between nodes until the convergence criteria are satisfied for the service and arrival rates of each node $i$. In case of closed networks, there is an additional fixed population constraint; that is, $\sum_{i=1}^{N} L_i = K$ (in closed queueing networks, the sum of mean queue lengths should add up to $K$). Furthermore, in this case it is not possible to obtain the exact values of the $\beta_i$'s. The authors approached this problem by first constructing a pseudo open network that does not have external arrivals and satisfies the fixed population constraint. All other parameters of the open network are the same as those for the closed network under consideration. The $\beta_i$'s obtained by solving the pseudo open network are used as initial values in the approximation for the closed network. The main steps of the algorithm are given as follows.

*Algorithm 9*

**S0:** Initialize $c_{di}^2$ and $c_{ai}^2$ for each $i = 1, \ldots, N$. Furthermore, let for some node $m$, $\lambda_m$ be initialized to an initial value.

**S1:** Solve $\lambda_i = \sum_{j=1}^{N} p_{ji}\lambda_j, \quad i = 1, \ldots, N$

**S2:** Solve the above optimization problem together with the fixed population constraint and obtain $\beta_i$, $i = 1, 2, 3$, and $\lambda_m$. Then obtain the marginal queue length probabilities.

**S3:** Obtain the new values of $c_{di}^2$ and $c_{ai}^2$. If the new values are not close to the previous values, then stop.

Goto **S1**.

**S4:** Implement the product form solution to calculate the throughputs and the mean queue lengths of nodes. Iterate until the job flow equations are satisfied

Validation tests given in the cited paper show that the algorithm is fairly accurate. No discussion on the time complexity of the algorithm is given. The algorithm may be applicable to other types of blocking once the principles of the maximum entropy is understood. The main difficulty of applying the concept is obtaining the behavior of a node in isolation and describing the relationships between nodes. Finally, we note that this is the only algorithm reported in the literature to analyze closed networks under RS-RD blocking with arbitrary topologies and general service times.

## 4. SYMMETRIC NETWORKS

In this section we discuss the concept of indistinguishable nodes as introduced by Onvural [1987] and Persone and Grillo [1987] in symmetric cyclic networks. When applicable, this notion allows the solution of the rate matrix of such networks on a reduced state space. It can be used as an efficient method to validate approximations as well as to study systems with symmetric parameters.

Consider an exponential cyclic network under BAS blocking with parameter $B_i = B$ and $\mu_i = \mu$, $i = 1, \ldots, N$. The algorithm presented next uses an aggregate state space obtained from the original state space after it is reduced by a factor of $N$.

Consider this cyclic network under BAS blocking with $B = 2$, $K = 4$, and $N = 3$ shown in Figure 22. The state space of this network has the following structure with all transition rates being equal to $\mu$.

Let $P(i, j, k)$ be the steady-state queue length probability of having $i$, $j$, and $k$ customers at nodes 1, 2, and 3, respectively. Furthermore, we used $I_i = B + 1$ (= 3) to denote that node $i$ is blocking the preceding node. Writing down the global balance equations from the above transition rate diagram and solving the system of equations numerically with the normalization equation replacing one of the equations, 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 since the 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:

$$S_1 = \{(2, 2, 0), (0, 2, 2), (2, 0, 2)\}$$
$$S_2 = \{(2, 3, 0), (0, 2, 3), (3, 0, 2)\}$$
$$S_3 = \{(2, 1, 1), (1, 2, 1), (1, 1, 2)\}$$
$$S_4 = \{(3, 1, 1), (1, 3, 1), (1, 1, 3)\}$$

Then, we have the state space structure for these equivalence classes with all transition rates being equal to $\mu$ as shown in Figure 23.

Writing down the global balance equations from the above transition rate diagram and solving the system of equations numerically with the normalization equation replacing one of the equations, we have

$$P(S_1) = 0.214287;$$
$$P(S_2) = 0.37515;$$
$$P(S_3) = 0.28571;$$
$$P(S_4) = 0.14285.$$

Furthermore, $P(S_i) = \sum_{(i_1,i_2,i_3)\in S_i} P(i_1, i_2, i_3)$, $i = 1, \ldots, 4$. 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 numerically. Algorithm 11 summarizes this procedure to obtain the queue length probabilities of the original network.

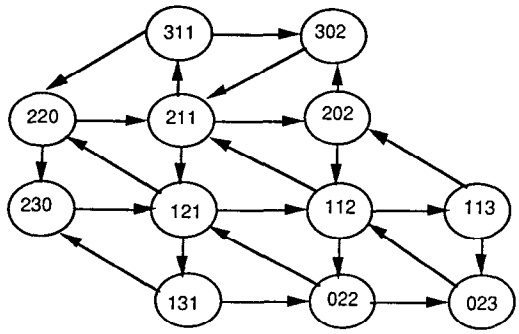


**Figure 22.** Transition rate diagram of a symmetric cyclic network under BAS blocking.

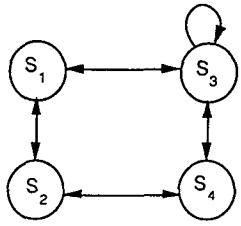


**Figure 23.** Transition rate diagram of the aggregated network.

*Algorithm 10*

**S1:** Generate the equivalence classes $S_i$, and set up the rate matrix.

**S2:** Solve the system numerically to obtain $P(S_i)$.

**S3:** 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 the equivalence class $i$.

**S4:** $P(i_1, i_2, \ldots, i_N) = G_K^{-1} P(S_i)$.

Finally, we note that although the concept of indistinguishable nodes is discussed in cyclic networks under BAS blocking, it

is also applicable to other blocking mechanisms defined in Section 1 in cyclic exponential cyclic networks and the central server model.

## 5. CONCLUSIONS

In this paper we give a survey of analytical, approximate, and numerical results related to closed queueing networks with finite queues. Except for a few special cases, these networks could not be shown to have product form solutions. Although the steady-state queue length distributions of these networks can, in theory, be calculated by solving the global balance equations together with the normalization equation numerically, this procedure can, in practice, be restrictive due to the time complexity of the procedure and the large storage required to store the rate matrixes, particularly for large networks. Since exact values of their steady-state queue length distributions are, in general, not attainable, good approximation algorithms are required to analyze closed queueing networks with finite queues.

Approximations developed in the literature (particularly for BAS and BBS blocking mechanisms) are, in general, based on empirical observations. In view of this, there is a strong need for developing approximation procedures to analyze closed networks with blocking that cannot otherwise be analyzed. One approach toward the development of approximation algorithms is decomposing the network into individual queues and analyzing them in isolation. This methodology has been used in numerous algorithms developed for open networks with blocking [Perros 1989]. Its extension to closed networks, however, is trivial. Exact decomposition of queueing networks with blocking (open or closed) requires state dependent arrival and service rates [Onvural 1989a]. This dependency on the number of customers appears not to be crucial in the case of open networks as a number of approximations has been developed with state independent parameters. Due to the fixed population of customers, however, the dependencies of the parameters of a node in isolation appears to be very strong for closed networks. Hence, the next generation of approximation algorithms should focus on obtaining good estimates of these state dependent parameters. Since this approach is based on the exact decomposition, such studies would produce efficient and accurate tools for the analysis of closed queueing networks with blocking.

## ACKNOWLEDGEMENTS

I am indebted to Professor Harry G. Perros for introducing the concept of blocking to me. During my studies, he not only motivated and encouraged my work but also created the environment that made it possible. I also would like to acknowledge the patience and assistance of Professor Richard R. Muntz whose knowledge in the field and numerous constructive suggestions helped me organize the paper and improve its clarity. Finally, I would like to thank Ms. Kelly Blasko, Professor Ian Akyildiz, Professor Salvatore March, and the reviewers for their suggestions and improvements.

## REFERENCES

AKYILDIZ, I. F. 1987. Exact product form solutions for queueing networks with blocking. *IEEE Trans. Comput. 1,* 121–126.

ALYILDIZ, I. F. 1988a. *General Closed Queueing Networks with Blocking*, Performance '87, Courtois and Latouche, eds. Elsevier North Holland, Amsterdam, 282–303.

AKYILDIZ, I. F. 1988b. On the exact and approximate throughput analysis of closed queueing networks with blocking. *IEEE Trans. Softw. Eng. SE-14,* 62–71.

AKYILDIZ, I. F. 1988c. Mean value analysis for blocking queueing networks. *IEEE Trans. Softw. Eng. SE-14* (1), 418–429.

AKYILDIZ, I. F. 1989a. Product form approximations for queueing networks with multiple servers and blocking. *IEEE Trans. Comput. 38,* 99–115.

AKYILDIZ, I. F. 1989b. Analysis of queueing networks with rejection blocking. In *First International Workshop on Queueing Networks with Blocking*, Perros and Altiok, eds. Elsevier North Holland, Amsterdam.

AKYILDIZ, I. F., AND LIEBEHERR, J. 1989. Application of Norton's theorem on queueing networks with finite capacities. In the *Proceedings of INFOCOM 89*, pp. 914–923.

AKYILDIZ, I. F., AND VON BRAND, H. 1990. Dual and selfdual networks of queues with rejection blocking. *Comput. J.* To appear.

AKYILDIZ, I. F., AND VON BRAND, H 1989a. Exact solutions for open, closed and mixed queueing networks with rejection blocking. *Theor. Comput. Sci. J. 64,* 203–219.

AKYILDIZ, I. F., AND VON BRAND, H. 1989b. Computation of performance measures for open, closed and mixed networks with rejection blocking. *Acta Info. 26*, 559–576.

AKYILDIZ, I. F., AND VON BRAND, H. 1989c. Central server models with multiple job classes, state dependent routing and rejection blocking. *IEEE Trans. Softw. Eng.* To appear.

ALTIOK, T., AND PERROS, H. G. 1987. Approximate analysis of arbitrary configurations of queueing networks with blocking. *Ann. OR 9*, 481–509.

AMMAR, M. H., AND GERSHWIN, S. B. 1987. Equivalence relations in queueing models of assembly/disassembly networks. Working paper. Georgia Institute of Technology.

BALSAMO, S., AND DONATIELLO, L. 1988. Two-stage cyclic network with blocking: Cycle time distribution and equivalence properties. In the *Proceedings of Modeling Techniques and Tools for Computer Performance Evaluation.* Potier and Puigjaner, eds. pp. 513–528.

BALSAMO, S., AND IAZEOLLA, G. 1983. Some equivalence properties for queueing networks with and without blocking. In *Performance '83*, Agrawala and Tripathi, eds. North-Holland Publishing Company, Amsterdam, pp. 351–360.

BALSAMO, S., V. DE NITTO PERSONE, AND IAZEOLLA, G. 1986. Some equivalencies of blocking mechanisms in queueing networks with finite capacity. Manuscript, Dipartimento di Informatica, Universite di Pisa, Italy.

BASKETT, F., CHANDY, K. M., MUNTZ, R. R., AND PALACIOS, F. G. 1975. Open, closed and mixed networks of queues with different classes of customers. *J. ACM 22*, 2, 249–260.

BOXMA, O., AND KONHEIM, A. 1981. Approximate analysis of exponential queueing systems with blocking. *Acta Info. 15*, 19–66.

CASEAU, P., AND PUJOLLE, G. 1979. Throughput capacity of a sequence of transfer lines with blocking due to finite waiting room. *IEEE Trans. Softw. Eng. 5*, 631–642.

CHANDY, K. M., AND SAUER, C. H. 1978. Approximate methods for analyzing queueing network models of computing systems. *ACM Comput. Surv. 10*, 3, 281–317.

CHANDY, K. M., HERZOG, U., AND WOO, L. 1975. Parametric analysis of queueing networks. *IBM J. Res. Dev. 19*, 43–49.

COFFMAN, E. G., ELPHICK, M. J., AND SHOSHANI, A. 1971. System deadlocks. *ACM Comput. Surv. 2*, *4*, 67–78.

COHEN, J. W. 1969. *The Single Server Queue.* North Holland Publishing Company, Amsterdam.

COURTOIS, P. J. 1977. *Decomposability: Queueing and Computer System Applications.* Academic Press, New York.

COX, D. R. 1955. A use of complex probabilities in the theory of stochastic processes. In the *Proceedings of the Cambridge Philosophical Society*, vol. 51, pp. 313–319.

DALLERY, Y., AND FREIN, Y. 1989. A decomposition method for the approximate analysis of closed queueing networks with blocking. In the *Proceedings of the 1st International Workshop on Queueing Networks with Blocking*, Perros and Altiok, eds. Elsevier North Holland, Amsterdam.

DALLERY, Y., AND YAO, D. D. 1986. Modeling a system of flexible manufacturing cells. In *Modeling and Design of Flexible Manufacturing Systems*, Kusiak, ed. Elsevier North Holland, Amsterdam, pp. 289–300.

DIEHL, G. W. 1984. A buffer equivalency decomposition approach to finite buffer queueing networks. Ph.D. dissertation. Eng. Sci., Harvard Univ.

GERSHWIN, S., AND BERMAN, U. 1981. Analysis of transfer lines consisting of two unreliable machines with random processing times and finite storage buffers. *AIIE Trans. 13*, (1), 2–11.

GORDON, W. J., AND NEWELL, G. F. 1967a. Cyclic queueing systems with restricted queues. *Oper. Res. 15*, 266–278.

GORDON, W. J., AND NEWELL, G. F. 1967b. Closed queueing systems with exponential servers. *Oper. Res. 15*, 254–265.

GROSS, D., AND HARRIS, C. M. 1974. *Fundamentals of Queueing Theory.* John Wiley, New York.

HIGHLEYMAN, W. H. 1989. *Performance Analysis of Transaction Processing Systems.* Prentice Hall, Englewood Cliffs, N.J.

HORDIJK, A., AND VAN DIJK, N. 1981. Networks of queues with blocking. In *Performance '81.* Klystra, ed. North Holland, Amsterdam, pp. 51–65.

HORDIJK, A., AND VAN DIJK, N. 1982. Adjoint processes, job local balance and insensitivity of stochastic networks. Bull:44 session. *Int. Stat. Inst. 50*, 776–788.

HORDIJK, A., AND VAN DIJK, N. 1983. Networks of queues: Part I—Job local balance and the adjoint process; Part II—General routing and service characteristics. In the *Proceedings International Conference Modeling Computing Systems*, Vol. 60, pp. 158–205.

JACKSON, J. R. 1963. Jobshop-like queueing systems. *Manage. Sci. 10* (1), 131–142.

JENNINGS, A. 1977. *Matrix Computation for Engineers and Scientists.* John Wiley, New York.

JUN, K. P. 1988. Approximate analysis of open queueing networks with blocking. Ph.D. dissertation, Operations Research Program, North Carolina State Univ.

KELLY, K. P. 1979. *Reversibility and Stochastic Networks.* John Wiley, Chichester, England.

KENDALL, D. G. 1953. Stochastic processes occurring in the theory of queues and their analysis by the method of imbedded Markov chains. *Ann. Math. Statist. 24*, 338–354.

KLEINROCK, L. 1976. *Queueing Systems.* Vol. II. John Wiley, New York.

KOUVATSOS, D. D. 1983. Maximum entropy methods for general queueing networks. Tech. Rep. RCC34, Univ. Bradford, U.K.

KOUVATSOS, D. D., AND XENIOS, N. P. 1989. Maximum entropy analysis of general queueing networks with blocking. *First International Workshop on Queueing Networks with Blocking,* Perros and Altiok, eds. North Holland, Amsterdam.

KUNDU, S., AND AKYILDIZ, I. F. 1989. Deadlock free buffer allocation in closed queueing networks. *Queue. Syst. J. 4,* 47–56.

LAW, A. M., AND KELTON, W. D. 1982. *Simulation Modeling and Analysis.* McGraw Hill, New York.

LITTLE, J. D. C. 1961. A proof of the queueing formula $L = \lambda W$. *Oper. Res. 9,* 383–387.

MARIE, R. 1979. An approximate analytical method for general queueing networks. *IEEE Trans. Softw. Eng. SE-5* (5), 530–538.

MINOURA, T. 1982. Deadlock avoidance revisited. *J. ACM 29* (4), 1023–1048.

MITRA, D., AND MITRANI, I. 1988. Analysis of a novel discipline for cell coordination in production lines. AT&T Bell Labs Res. Rep.

MUNTZ, R. R. 1978. Queueing networks: A critique of the state of the art directions for the future. *ACM Comput. Surv. 10,* 3, 353–359.

ONVURAL, R. O. 1987. Closed queueing networks with finite buffers. Ph.D. dissertation, CSE/OR, North Carolina State Univ.

ONVURAL, R. O. 1989a. On the exact decomposition of exponential closed queueing networks with blocking. *First International Workshop on Queueing Networks with Blocking,* Perros and Altiok, eds. North Holland, Amsterdam.

ONVURAL, R. O. 1989b. Some product form solutions of multi-class queueing networks with blocking. *Perform. Eval.* Special Issue on Queueing Networks with Blocking, Akyildiz and Perros, eds. *10–11,* 247–254.

ONVURAL, R. O., AND PERROS, H. G. 1986. On equivalencies of blocking mechanisms in queueing networks with blocking. *Oper. Res. Lett. 5,* (6), 293–298.

ONVURAL, R. O., AND PERROS, H. G. 1988. Equivalencies between open and closed queueing networks with finite buffers. International Seminar on the Performance Evaluation of Distributed and Parallel Systems, Kyoto, Japan. *Perform. Eval., 9,* 263–269.

ONVURAL, R. O., AND PERROS, H. G. 1989a. Some equivalencies on closed exponential queueing networks with blocking. *Perform. Eval. 9,* 111–118.

ONVURAL, R. O., AND PERROS, H. G. 1989b. Throughput analysis in cyclic queueing networks with blocking. *IEEE Trans. Softw. Eng. SE-15,* (6), 800–808.

PERROS, H. G. 1984. Queueing networks with blocking: A bibliography. *ACM Sigmetrics, Perf. Eval. Rev. 12,* (2), 8–12.

Received June 1987; final revision accepted April 1989.

PERROS, H. G. 1989. Open queueing networks with blocking. In *Stochastic Analysis of Computer and Communications Systems,* Takagi, ed. Elsevier North Holland, New York.

PERROS, H. G., NILSSON, A., AND LIU, Y. G. 1988. Approximate analysis of product form type queueing networks with blocking and deadlock. *Perform. Eval. 8,* 19–39.

PERSONE, DE NITTO, V., AND GRILLO, D. 1987. Managing blocking in finite capacity symmetrical ring networks. In the *3rd Conference on Data and Communication Systems and Their Performance,* Rio de Janeiro, Brazil.

PITTEL, B. 1979. Closed exponential networks of queues with saturation: The Jackson type stationary distribution and its asymptotic analysis. *Math. Oper. Res. 4,* 367–378.

REISER, M. 1979. A queueing network analysis of computer communications networks with window flow control. *IEEE Trans. Comm. 27,* 1199–1209.

SHANTHIKUMAR, G. J., AND YAO, D. D. 1989. Monotonicity properties in cyclic queueing networks with finite buffers. *First International Workshop on Queueing Networks with Blocking.* Perros and Altiok, eds. North Holland, Amsterdam.

SOLOMON, S. L. 1983. *Simulation of Waiting Line Systems.* Prentice Hall, Englewood Cliffs, N.J.

SURI, R., AND DIEHL, G. W. 1984. A new building block for performance evaluation of queueing networks with finite buffers. In the *Proceedings of the ACM Sigmetrics on Measurement and Modeling of Computer Systems,* 134–142.

SURI, R., AND DIEHL, G. W. 1986. A variable buffer size model and its use in analytical closed queueing networks with blocking. *Manage. Sci. 32* (2), 206–225.

VAN DIJK, N. M., AND TIJMS, H. C. 1986. *Insensitivity to Two Node Blocking Models with Applications, Teletraffic Analysis and Computer Performance Evaluation.* Boxma, Cohen, and Tijms, eds. Elsevier North Holland, Amsterdam, The Netherlands, pp. 329–340.

YAO, D. D., AND BUZACOTT, J. A. 1985a. Modeling a class of state dependent routing in flexible manufacturing systems. *Ann. OR 3,* 153–167.

YAO, D. D., AND BUZACOTT, J. A. 1985b. Queueing models for flexible machining station Part I: Diffusion approximation. *Eur. J. Oper. Res. 19,* 233–240.

YAO, D. D., AND BUZACOTT, J. A. 1985c. Queueing models for flexible machining stations Part II: The method of Coxian phases. *Eur. J. Oper. Res. 19,* 241–252.

YAO, D. D., AND BUZACOTT, J. A. 1986. The exponentialization approach to flexible manufacturing system models with general processing times. *Eur. J. Oper. Res. 24,* 410–416.