Mean Value Analysis for Blocking Queueing Networks

IAN F. AKYILDIZ, MEMBER, IEEE

Abstract—**Mean value analysis is an exact solution technique for infinite capacity queueing networks and enjoyed widespread popularity during recent years. It considers the behavior of the system by stepwise increasing the number of jobs in the entire network, thus it is well-suited for the analysis of queueing networks with blocking. In this work, an approximation is introduced for the mean value analysis of queueing networks with transfer blocking. The blocking occurs when a job after completing service at a station wants to join a station which is full. The job resides in the server of the source station until a place becomes available in the destination station. The approximation is based on the modification of mean residence times due to the blocking events that occur in the network. Several examples are executed in order to validate the approximate results.**

Index Terms—**Blocking, finite station capacities, performance evaluation, performance measures, queueing network models.**

I. Introduction

QUEUEING networks have increased their importance in performance evaluation of computer systems and communication networks in the last two decades. When analyzing systems with infinite station capacities, several methods have been introduced in the last 15 years. In a major work, Baskett, Chandy, Muntz, and Palacios [8] have shown that queueing networks with particular station types ($M/M/m_i$-FCFS, M/G/1-RR-PS, M/G/Infinite Servers and M/G/1-LCFS-PR) have a product form solution. In these solutions, the equilibrium state probability of the network can be expressed as a product of terms for each station in the network. The product form solution implies each station in the network can be analyzed independently. Several different algorithms have been proposed for product form networks. One of the methods which has attracted particular interest is the mean value analysis of Reiser and Lavenberg [16]. Mean value analysis is based on three fundamental formulas for computing the mean residence time, throughput, and the mean number of jobs in the stations. It performs an iterative computation of the desired performance measures. The principal advantage of mean value analysis is that it is very fast and easy to implement.

In actual computer systems, resources have finite capacity. Thus queueing networks with blocking should be used to investigate them analytically. A queueing network with blocking can be regarded as a collection of stations with *finite* capacities through which jobs proceed in order to satisfy their service requirements. Blocking occurs due to the finite capacities of the individual stations.

We consider the transfer blocking type where blocking occurs when a job completing service at station i cannot proceed to station j because station j is full. In this case station i's server stops processing until station j releases a job.

Since blocking causes interdependencies between stations, product form or other approximate methods for infinite capacity networks cannot be applied in their original forms. Simulation and/or numerical analysis is generally used instead. This introduces major problems, as simulation is expensive and statistically inaccurate while numerical methods are restricted to very small networks (since the state space grows rapidly with the number of stations and jobs).

In recent years there has been a growing interest in the development of computational methods to analyze blocking queueing networks. The interest developed primarily from the realization that these models are useful in the study of the subsystem behavior in computers and communication networks. In addition they provide detailed descriptions of several computer-related applications.

Various types of blocking have been reported in the literature so far. In the following we discuss those studies which investigate the transfer blocking type.

In [1] we show that the state spaces of a closed network with two stations and transfer blocking and the same network without blocking agree exactly if the number of jobs in the nonblocking network is adjusted properly. In [2] we extend this concept to networks with more than two stations. Here the state spaces cannot be made to agree exactly. An approximation is found by selecting the number of jobs in the nonblocking network so that the number of states are the same as in the blocking network. This method allows to compute throughputs. The results of extensive validations of the method are presented, and they indicate very good accuracy. In order to compute the mean number of jobs we introduce in [3] the so-called *state normalization technique* where we consider the blocking network simply as a product form network. In certain states the capacity restrictions of some stations will then be violated. The jobs that exceed the station capacity are dis-

Manuscript received August 30, 1985.
The author is with the School of Information and Computer Science, Georgia Institute of Technology, Atlanta, GA 30332.
IEEE Log Number 8718382.

0098-5589/88/0400-0418$01.00 © 1988 IEEE

tributed to the stations upstream according to the routing probabilities. This method uses a large amount of computation but it gives good accuracy.

Onvural and Perros [11] show that if the number of jobs in the network is one more than the capacity of the station with smallest capacity there is an exact product form solution. Essentially, what happens is that the blocked server functions as an additional space in the queue of the station that is blocking it. This is true since at most one server can be blocked at a time, so all the jobs (except the one in the blocked server) are in the blocking station. Viewed in this way, the network is nonblocking and can be solved by product form network algorithms.

Suri and Diehl [21] consider the transfer blocking policy in cyclic and tandem networks. They present an approximate method to compute the throughput of the network. If the network is a cyclic network, one of the stations has to be nonblocking. If the network is tandem, the arrival process must be Poisson. They approximate groups of two stations by a variable capacity station, defined as a superposition of fixed capacity stations. They start with the last two stations and successively reduce the network until two stations in tandem remain. The method is easy to implement and shows good accuracy but involves much computation. At each step all conditional probabilities have to be found, since they are used to construct the equivalent variable capacity station. The method only gives the throughput of the entire network, it does not give statistics for individual stations.

Brandwajn and Jow [9] consider tandem networks with transfer blocking. The idea is to consider pairs of stations, where the state of a station is supplemented with a status indicator. The status indicator tells if the station is blocked or not. For a pair of stations in the network it is then possible to write down the transition rates in terms of the states of the neighboring stations. The resulting equations are solved iteratively.

Altiok [5] solves tandem networks with transfer blocking approximately. The delays caused by blocking are represented by a phase distribution. The job at the front of the queue travels through phases that represent blocking delays after leaving the phase that represents its service requirement. The blocking delays are computed by starting at the last stations in the network and going upstream, considering each station as an M/G/1 station.

Schweitzer and Altiok [18] consider aggregation approximations for tandem networks with transfer blocking. Aggregate states that represent groups of states of the network are introduced. The balance equations for each station can then be written down in terms of parameters, like the probability that the downstream station is full and the state-dependent arrival rates, that depend on the rest of the network.

Note that deadlock is possible in transfer blocking networks. All stations in a directed cycle could be full at one time. If in each of the stations of the cycle the blocked job is scheduled to go to the next station in the cycle, the network is deadlocked. There are two possible solutions to the deadlock problem:

i) Include a strategy to handle deadlocks in the model. Perros, Nilsson, and Liu [15] assume that in case of deadlock *all* jobs involved move *simultaneously* to their destinations. This complicates the model, since the deadlock handling method influences the balance equations.

ii) Simply restrict yourself to cases where deadlock is impossible. One such case arises whenever the number of jobs in the system is less than the capacity of the directed cycle with minimal capacity. No directed cycle can ever have all its stations full at the same time, and deadlock is impossible (Akyildiz and Kundu [4]).

Several other investigators have published results on queueing networks with transfer blocking in recent years [6], [10], [14], [22]. The area is surveyed by Onvural [11].

II. Model Assumptions

We consider closed queueing network models with N single server stations and a total of K jobs. The service time at each station is exponentially distributed with rates μ_i for $i = 1, \cdots, N$. Each station capacity is restricted to a finite limit M_i for $i = 1, \cdots, N$. Simply stated, M_i is the capacity of the ith station where M_i = (queue capacity + the number of servers). Blocking between stations i and j occurs when j contains M_j jobs. In this case, station i's server stops processing until station j releases a job (i.e., a job in service at station j finishes). Once station i is no longer blocked, it resumes its exponential service.

Further, we assume that

$$K < \sum_{i=1}^{M} M_i. \tag{1}$$

The total number of jobs in the network may not exceed the sum of the individual station capacities in the network. The service discipline utilized at each station is first-come-first-served.

As mentioned before, this proposed model is the transfer blocking queueing network studied by several investigators in recent years [13]. Some years ago, Bard [7] proposed an approximation method based upon mean value analysis for queueing networks with blocking. However, in the validation studies, we found large inaccuracies in his approximation method.

In this work, we introduce another approximation of mean value analysis for blocking queueing networks. In the next section the algorithm is described in detail. In Section IV we give the summary of the algorithm. Section V contains a numerical example which explains the general flow of the algorithm. The last section covers the evaluation of the algorithm. The Appendix contains results of sample queueing networks with different topologies.

III. Mean Value Analysis for Blocking Queueing Networks (MVABLO)

MVABLO is based on the classical mean value analysis, MVA, of Reiser and Lavenberg [16]. MVA was developed from two major theorems: Arrival Instant Distribution Theorem [19] and Little's Law. From the first theorem, Reiser and Lavenberg [16] derived a formula for the *mean residence time* (time spent by a job in queue and in service) of a job in the ith station:

$$\bar{t}_i(k) = \frac{1}{\mu_i}[1 + \bar{k}_i(k-1)] \tag{1}$$

where $\bar{k}_i(k-1)$ is the mean number of jobs in the ith station assuming that there are $(k-1)$ jobs in the entire network. The informal interpretation of this formula is easy. The mean residence time of a job entering the ith station is given by its own mean service time plus the mean service time of all jobs which are already in the queue or in service at that station.

From the second theorem, the *throughput* of the network can easily be derived:

$$\lambda(k) = \frac{k}{\sum_{i=1}^{N} e_i \bar{t}_i(k)} \tag{2}$$

where e_i is the mean number of visits a job makes to station i and is given by:

$$e_i = \sum_{j=1}^{N} e_j p_{ji} \qquad \text{for } i = 1, \cdots, N$$

where p_{ji} is the probability that a job in the jth station proceeds to the ith station.

The *mean number of jobs* in the ith station can also be derived from the second theorem:

$$\bar{k}_i(k) = e_i \lambda(k)\, \bar{t}_i(k). \tag{3}$$

$\bar{k}_i(0) = 0$ is assumed for the initial value in the iterations. The iteration terminates when the total or desired number of jobs in the network is reached.

As mentioned before, mean value analysis can be applied only to product form networks of the form presented above. However, the stepwise behavior of the MVA permits the algorithmic determination of blocking events in networks with finite station capacities. Two basic characteristics of blocking network models must be considered in the algorithm:

i) A station whose successor station capacity is full is blocked.

ii) A station whose capacity is full cannot accept any job.

MVABLO starts with classical mean value analysis and computes the mean residence time from (1), throughput from (2), and the mean number of jobs in each station from (3). After each iteration, we check to guarantee that the mean number of jobs in each station is less than or equal to the capacity of that station, (i.e., $\bar{k}_i \leq M_i$). If this is not the case, we must consider the two additional characteristics of blocking networks mentioned above.

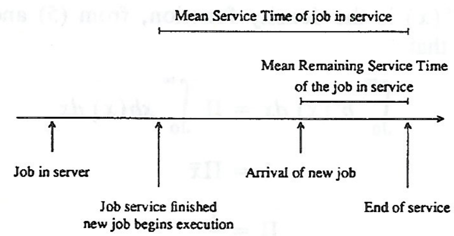


Fig. 1.

The fact that a job cannot join another station with a full capacity has the effect of increasing the mean residence time of the source station. The job blocks the source station until a place is available in the full destination station. This place will be available after a job has finished service at the full station. Accordingly, the mean residence time of the jobs in the blocked station increases by the mean remaining service time BT_i of the destination station:

$$\bar{t}_j(k) = \frac{1}{\mu_j}[1 + \bar{k}_j(k-1)] + BT_i$$

$$\text{for } i, j = 1, 2, \cdots, N \text{ and } i \neq j. \tag{4}$$

In order to determine BT_i we utilize the following known theorem [17].

Theorem: The mean remaining service time BT_i of the destination station is exactly its mean service time $1/\mu_i$.

Proof: We consider an M/G/1-FCFS system with $F_a(t) = 1 - \exp\{-\lambda t\}$ as the arrival distribution function and $F_s(t) = \textit{Arbitrary}$ as the service time distribution function.

The waiting time of an arriving job is computed by adding the mean remaining service time to the product of the queue length and the mean service time. The mean remaining service time is the time that the job in service still needs to complete at the moment the new job enters the system.

The mean remaining service time for a newly arriving job is zero if the server is inactive. In this case, the new job is served immediately. In our consideration this case is irrelevant, since the capacity of the station is already full whenever BT_i is to be computed. Therefore, we compute the mean remaining service time of a job assuming the system is active.

Let $b(x)$ be the density function of the service time $\bar{x} = 1/\mu$. Let $b'(x)$ be the density function of the service time intervals where a job enters the system.

$$b'(x) \sim x \tag{5}$$

The longer the time interval, the greater is the probability that a job arrives in the system.

$$b'(x) \sim b(x) \tag{6}$$

The greater the probability of a time interval x, the greater the probability that a job arrives in this interval.

Since $b'(x)$ is the density function, from (5) and (6) it follows that

$$\int_0^\infty b'(x)\,dx = \Pi \int_0^\infty x b(x)\,dx \tag{7}$$

$$1 = \Pi \bar{x} \tag{8}$$

$$\Pi = \frac{1}{\bar{x}} \tag{9}$$

where Π is the proportional factor.

Then it follows:

$$b'(x) = \frac{x}{\bar{x} b(x)}. \tag{10}$$

The mean length of the service time when a new job arrives can easily be computed by

$$\bar{y} = \int_0^\infty x b'(x)\,dx = \frac{1}{\bar{x}} \int_0^\infty x^2 b(x)\,dx = \frac{\overline{x^2}}{\bar{x}}. \tag{11}$$

The mean remaining service time that a job must wait, BT, is given by:

$$BT = \frac{\bar{y}}{2} = \frac{\overline{x^2}}{2\bar{x}}. \tag{12}$$

By substituting the first $\{\bar{x} = 1/\mu\}$ and second moments $\{\overline{x^2} = 2/\mu^2\}$ for the exponential distribution in (12), we obtain the following result for the mean remaining service time:

$$BT = \frac{1}{\mu}. \tag{13}$$

Thus the time that a blocked job must wait in the source station is exactly the mean service time of the destination station. This completes the proof.

If a source station has many successor stations and one of them is full, the mean residence time of the source station increases by the mean service time of the full station multiplied by the transition probability by which the job would proceed to the full station, weighted by the ratio of the mean number of visits of the full station (e_j) to the mean number of visits of the blocked station (e_i).

$$\bar{t}_j(k) = \frac{1}{\mu_j}\left[1 + \bar{k}_j(k-1)\right] + BT_i\left\{\frac{p_{ji} e_j}{e_i}\right\} \tag{14}$$

The second general characteristic of blocking networks is that a full station cannot accept a new job. As a result, the mean residence time is computed by the mean service times of jobs which are already in the station:

$$\bar{t}_i(k) = \frac{1}{\mu_i}\left[\bar{k}_i(k-1)\right]. \tag{15}$$

If the capacity of a station is exceeded in an iteration, that is, $\bar{k}_i(k) > M_i$, we repeat the iteration. New mean residence times are derived from (14) and (15). Throughput is computed using (2). Finally the mean number of jobs is computed via (3). In all following iterations, we use (14) and (15) for the computation of the mean residence times. If an additional blocking event occurs in the same station, the mean residence time of the destination station remains the same while the mean residence time of the blocked station is again increased.

IV. The Algorithm Summary

begin
  for all stations $i = 1$ to N **do**
    · $\bar{k}_i(0) = 0$
    · $BT_i(0) = 0$
    · $z_i(0) = 1$
    · compute e_i, the mean number of visits by a job to station i
  end
  for all jobs in the network $k = 1$ to K **do**
  begin
    Repeat
    for all stations $i = 1$ to N **do**
      1. $\bar{t}_i(k) = \frac{1}{\mu_i}[z_i(k) + \bar{k}_i(k-1)] + BT_i(k)$
    end
    2. $\lambda(k) = \dfrac{k}{\sum_{i=1}^{N} e_i \bar{t}_i(k)}$
    for all stations $i = 1$ to N **do**
      3. $\bar{k}_i(k) = \bar{t}_i(k)\, e_i \lambda(k)$
    end
    4. **if** there exists a $\bar{k}_i(k) > M_i$ **then**
      · $z_i(k) := 0$
      · $BT_j(k) := BT_j(k) + \frac{1}{\mu_i}\left\{\frac{p_{ji} e_j}{e_i}\right\}$
    else
      · $z_i(k+1) := z_i(k)$
      $BT_i(k+1) := BT_i(k)$
    endif
    until $\bar{k}_i(k) < M_i$
  end {for $k = 1$ to K}
end

We can also compute other interesting performance measures using the MVABLO results. The *blocking probability* of a station is computed from the proportion of the mean blocking time to the mean residence time of jobs in a station:

$$P_{B_i}(k) = \frac{BT_i(k)}{\bar{t}_i(k)} \quad \text{for } i = 1, 2, \cdots, N. \tag{16}$$

The *mean waiting time* of a job in the ith station is computed by subtracting the mean service time and the mean blocking time of jobs from the mean residence time:

$$\bar{w}_i(k) = \bar{t}_i(k) - \frac{1}{\mu_i} - BT_i(k) \quad \text{for } i = 1, 2, \cdots, N. \tag{17}$$

The *throughput* of each station is given by:

$$\lambda_i(k) = e_i \lambda(k) \quad \text{for } i = 1, 2, \cdots, N. \tag{18}$$

The *utilization* is defined as:

$$\rho_i(k) = \frac{\lambda_i(k)}{\mu_i} \quad \text{for } i = 1, \cdots, N. \quad (19)$$

The mean queue length in the ith station can be obtained by Little's Law:

$$\overline{Q}_i(k) = \overline{w}_i(k)\,\lambda_i(k) \quad \text{for } i = 1, \cdots, N. \quad (20)$$

In the following numerical example, we outline the general flow of the MVABLO algorithm.

V. Numerical Example

Examine a closed queueing network model with $N = 3$ stations and $K = 27$ jobs. The stations have finite capacities $M_1 = 12$, $M_2 = 10$, and $M_3 = 14$. The service time distribution is exponential with the mean rates $\mu_1 = 1$, $\mu_2 = 2$, and $\mu_3 = 3$. The stations are serially connected.

Until the iteration ($k = 14$) the MVABLO behaves exactly like classical MVA for networks with infinite station capacities. The mean number of jobs $\overline{k}_i$ never exceeds the capacity limit of the respective stations. For $k = 14$ we obtain the following values for the mean number of jobs in each station:

$$\overline{k}_1(14) = 12.507 \quad \overline{k}_2(14) = 0.9992 \quad \overline{k}_3(14) = 0.4999.$$

Now the mean number of jobs in the first station violates the capacity limit,

$$\overline{k}_1(14) = 12.5017 > M_1 = 12.$$

As a result, we have to modify the mean residence times. First the modifications in step 4 of the algorithm are performed.

$$z_1(14) = 0 \quad BT_1(14) = 0 \quad BT_2(14) = 0 \quad BT_3(14) = 1$$

We repeat step 1 and obtain the new mean residence times:

$$\overline{t}_1(14) = 11.503 \quad \overline{t}_2(14) = 0.999 \quad \overline{t}_3(14) = 1.5.$$

Step 2 yields

$$\lambda(14) = 0.999.$$

Step 3 obtains the new mean number of jobs in each station:

$$\overline{k}_1(14) = 11.502 \quad \overline{k}_2(14) = 0.999 \quad \overline{k}_3(14) = 1.49.$$

Obviously no capacity limitation is violated so that we can increase the iteration step. Continuing through the iterations, for $k = 26$ we have

$$\overline{k}_1(26) = 11.80 \quad z_1(26) = 0 \quad BT_1(26) = 0$$
$$\overline{k}_2(26) = 0.99 \quad z_2(26) = 1 \quad BT_2(26) = 0$$
$$\overline{k}_3(26) = 13.20 \quad z_3(26) = 0 \quad BT_3(26) = 9.$$

For $k = 27$ we compute

$$\overline{t}_1(27) = 11.80 \quad \lambda(27) = 1.017 \quad \overline{k}_1(27) = 12.01$$
$$\overline{t}_2(27) = 0.99 \qquad \overline{k}_2(27) = 1.02$$
$$\overline{t}_3(27) = 13.73 \qquad \overline{k}_3(27) = 13.97.$$

Since the first station is again full $\{\overline{k}_1(27) = 12.0108 > M_1 = 12\}$, the third station is blocked. Thus we have

$$BT_3(27) = BT_3(27) + \frac{1}{\mu_1}\left\{\frac{e_3 p_{31}}{e_1}\right\} = 10.$$

We now repeat steps 1–3 to obtain

$$\overline{t}_1(27) = 11.80 \quad \lambda(27) = 0.98 \quad \overline{k}_1(27) = 11.57$$
$$\overline{t}_2(27) = 0.99 \qquad \overline{k}_2(27) = 0.97$$
$$\overline{t}_3(27) = 14.73 \qquad \overline{k}_3(27) = 14.44.$$

Now the mean number of jobs in the third station violates the capacity limit, since

$$\overline{k}_3(27) = 14.4475 > M_3 = 14.$$

Therefore we modify

$$z_3(27) = 0 \quad BT_1(27) = 0$$
$$BT_2(27) = 0.333 \quad BT_3(27) = 10.$$

The mean residence times are obtained by repeating steps 1–3:

$$\overline{t}_1(27) = 11.80 \quad \lambda(27) = 0.9806 \quad \overline{k}_1(27) = 11.57$$
$$\overline{t}_2(27) = 1.33 \qquad \overline{k}_2(27) = 1.30$$
$$\overline{t}_3(27) = 14.40 \qquad \overline{k}_3(27) = 14.12.$$

Since the third station is still full $\{\overline{k}_3(27) > M_3 = 14\}$; it follows that the second station is again blocked:

$$BT_2(27) = BT_2(27) + \frac{1}{\mu_3}\left\{\frac{e_2 p_{23}}{e_3}\right\} = 0.666.$$

We repeat the iteration for $k = 27$ for the third time by returning to step 1 and obtain

$$\overline{t}_1(27) = 11.80 \quad \lambda(27) = 0.968 \quad \overline{k}_1(27) = 11.43$$
$$\overline{t}_2(27) = 1.66 \qquad \overline{k}_2(27) = 1.61$$
$$\overline{t}_3(27) = 14.4 \qquad \overline{k}_3(27) = 13.95.$$

Since no capacity limitations are violated and the total number of jobs in the network ($K = 27$) is reached, the last values shown above are the final results for performance measures. In Table I we list the MVABLO results and the exact results for this model which were obtained by numerical analysis [20].

VI. Evaluation

About 150 queueing network models with blocking were investigated for the validation of the method. The number of jobs was varied from 5 to 100 and the number of stations varied from 2 to 6. The analytical results were compared with simulation results obtained by the RESQ programming package [17] and with numerical analysis results [20].

The major advantage of the MVABLO is its extremely

TABLE I

	Exact	MVABLO	Deviation
$\bar{r}_1$	11.837	11.804	0.3
$\bar{r}_2$	1.622	1.664	2.6
$\bar{r}_3$	13.548	14.400	6.3
$\bar{k}_1$	11.830	11.436	3.3
$\bar{k}_2$	1.621	1.612	0.5
$\bar{k}_3$	13.544	13.952	2.9
λ	0.999	0.968	3.1

fast execution time. MVABLO also has very low storage requirement, on the order of $O[N(K + 1)]$.

Our validation study has shown that on the average MVABLO results differ by 10 percent from the simulation or exact numerical results. One reason for larger deviations is the checking of the station capacity violation. After each iteration we check whether or not the mean number of jobs is less than the capacity of the station. If this value is minimally less than the capacity of that station, we continue the iterations. But if the station capacity is minimally exceeded, then we modify the mean residence times. However, modifying mean residence times can introduce large deviations into a situation that originally had a minimum deviation. In such a case, the original mean residence times could be a better estimate than the modified results. Furthermore, the checking of station capacity violations does not always detect the blocking. Blocking might occur in reality even though the mean number of jobs does not exceed the capacity of the station in the iteration of MVABLO. In such a situation no modifications are performed on the mean residence times. MVABLO behaves exactly like the classical MVA in such cases. This can be observed in the examples given in the Appendix: 6 with 10 jobs, 7 with 15 jobs, and 8 with 10 jobs where MVABLO results agree with the classical MVA results.

Another cause of the large deviations results from the mean number of jobs in a station $\bar{k}_i(k)$ exceeding the capacity M_i, due to the increase in its blocking time. If the capacity is exceeded, we must lower the throughput so that the mean number of jobs in the full station will be less than the capacity. We will explore this in detail by considering the numerical example presented in Section V.

We have seen that the mean number of jobs in the third station $\bar{k}_3(27)$ exceeds the capacity M_3 for the first time during the iteration 27. The capacity is also violated after the first correction to the 27th iteration. This second violation occurs because of the increasing blocking time of the third station and not because of new arrivals. We obtained the mean number of jobs in the third station $\bar{k}_3(27)$ = 14.4475 and the mean residence time $\bar{r}_3(27) = 14.733$. Since the capacity of the third station is only $M_3 = 14$, we must lower the $\bar{k}_3(27)$ value.

First we set the counter $z_3(27)$ to zero. By setting the counter $z_3(27)$ to zero we can partly lower the $\bar{k}_3(27)$ value. The third station's mean service time $(1/\mu_3) = 0.333$ is smaller than the mean service time of the first station's mean service time $(1/\mu_1) = 1$. This is added to the mean residence time of the third station if the first station is full. So we obtain the mean residence time $\bar{r}_3(27) = 14.4$. As a result, the throughput value must be lowered at least to $\lambda(27) = 0.9722$ so that the $\bar{k}_3(27)$ value will be less than the capacity $M_3 = 14$. By comparing this throughput value $\lambda(27) = 0.9722$ to the exact value we establish a deviation of 2.7 percent. This continues through the next iterations.

We have concluded that if the mean service times of the stations are very different from each other, considerable capacity exceeding may occur in the model. These exceedings cannot be sufficiently lowered by merely setting the counter value $z_i(k) = 0$. This decreasing of the throughput value also leads to frequent blocking events at the station and thus to a large increase of the mean residence time of the predecessor stations. Hence deviations can occur as shown in the following tables. The tables illustrate that the increase of the blocking time is the second station causes the jobs to be removed from the first to the second station. The mean residence time of the first station will be decreased accordingly.

For K = 28:

	Exact	MVABLO	Deviation
$\bar{r}_1$	11.842	11.436	3.4
$\bar{r}_2$	2.619	3.306	26.2
$\bar{r}_3$	13.556	14.650	8.1
λ	0.999	0.953	4.7

For K = 29:

	Exact	MVABLO	Deviation
$\bar{r}_1$	11.849	10.894	8.1
$\bar{r}_2$	3.617	5.075	40.3
$\bar{r}_3$	13.568	14.652	8.0
λ	0.998	0.947	5.2

For K = 30:

	Exact	MVABLO	Dev.
$\bar{r}_1$	11.866	10.317	13.1
$\bar{r}_2$	4.613	6.569	42.4
$\bar{r}_3$	13.59	14.625	7.5
λ	0.997	0.952	4.6

For K = 32:

	Exact	MVABLO	Dev.
$\bar{r}_1$	11.965	9.391	21.5
$\bar{r}_2$	6.616	9.638	45.7
$\bar{r}_3$	13.738	14.666	6.7
λ	0.990	0.949	4.1

In most of the examples MVABLO exhibited behavior similar to the example shown above. In some iterations the results for performance measures were very close to the exact values. In some iterations the approximation could not capture the blocking events. However, the important fact to realize is that the MVABLO results are inherently correct.

MVABLO does not handle the deadlock problem. Whenever a deadlock occurs in a network MVABLO terminates after looping on one iteration for a predetermined number of repetitions, issuing a message to the effect that the system is deadlocked. In our implementation the repetition limit was set to 100.

For further investigation MVABLO could be improved in an attempt to remove the above mentioned problems. It could also be extended to networks with multiple server stations.

Appendix

We give results for 15 blocking queueing networks with various number of jobs. We have a total of 33 test cases with different system input parameters. The deviations δ are computed using the following relative formula:

$$\delta = \frac{|\text{Simulation Value} - \text{MVABLO Value}|}{\text{Simulation Value}} * 100.$$

The first 5 models are two-station-networks where the input parameters are:

Example	K	M_1	M_2	$1/\mu_1$	$1/\mu_2$
1	20	18	10	2	3.33
2	25	18	13	0.8	0.5
3	15	10	10	1.111	1.5
4	8	4	7	10	2.5
5	50	31	24	4	2

The MVABLO results are shown in the following table:

	$\bar{k}_1$			$\bar{k}_2$		
Example	Exact	MVABLO	δ	Exact	MVABLO	δ
1	10.38	11.02	6.3	8.98	9.62	6.4
2	17.12	17.57	2.7	7.88	7.42	5.8
3	6.37	5.22	22.3	8.63	9.77	7.4
4	3.92	3.91	3.1	4.07	4.09	2.4
5	30.57	30.41	1.6	19.43	19.58	0.4

	$\bar{r}_1$			$\bar{r}_2$			λ		
Example	Exact	MVABLO	δ	Exact	MVABLO	δ	Exact	MVABLO	δ
1	36.58	34.27	5.8	29.84	31.76	7.1	0.30	0.31	0.5
2	13.85	14.23	2.7	6.38	6.01	5.8	1.23	1.23	0
3	9.80	7.61	18	13.27	14.24	13.3	0.65	0.68	5.1
4	39.82	38.56	0.4	41.34	40.35	0.4	0.09	0.10	2.7
5	124.40	122.37	0.5	79.16	78.80	0.8	0.24	0.24	0

In the following tables we list results of arbitrary chosen numerical examples with 3, 4, and 5 stations. Each table contains three computations for various number of jobs. These are a) MVABLO, b) simulation, and c) NOCAP results. NOCAP implies that the network is analyzed by ignoring the blocking, i.e., setting the station capacities $M_i = \infty$, and using the standard product form solution.

The third column in each section contains the standard deviation of simulation results. $\delta 1$ shows the relative deviations between MVABLO and the simulation. $\delta 2$ shows the deviations between NOCAP and simulation. This column demonstrates the effects of finite station capacity on the performance of the network.

Example 1

$N = 3$ stations, (serially switched); $M_1 = 12$, $M_2 = 14$, $M_3 = 8$, $1/\mu_1 = 1.0$, $1/\mu_2 = 0.5$, $1/\mu_3 = 0.333$

i) $K = 15$ jobs:

	MVABLO	Simulation	Std. Dev.	δ_1(%)	NOCAP	δ_2(%)
$\bar{r}_1$	11.501	11.85	0.2	3.0	13.501	13.9
$\bar{r}_2$	0.999	0.96	4.9	3.4	0.999	3.3
$\bar{r}_3$	2.231	2.35	1.1	20.1	0.499	78.8
$\bar{k}_1$	11.25	11.71	0.2	3.9	13.500	15.2
$\bar{k}_2$	0.977	0.95	4.9	2.3	0.999	4.4
$\bar{k}_3$	2.770	2.33	1.1	18.7	0.499	78.6
λ	0.9783	0.988	1.1	1	0.999	1.1

ii) $K = 20$ jobs:

	MVABLO	Simulation	Std. Dev.	δ_1(%)	NOCAP	δ_2(%)
$\bar{r}_1$	11.793	11.970	0.1	1.47	18.501	54.5
$\bar{r}_2$	0.994	1.027	5.0	3.21	0.999	2.7
$\bar{r}_3$	7.212	7.224	0.6	0.16	0.499	93
$\bar{k}_1$	11.985	11.840	0.1	1.20	18.500	56.2
$\bar{k}_2$	1.010	1.016	5	0.49	0.999	1.6
$\bar{k}_3$	7.332	7.144	0.6	2.60	0.499	93
λ	0.984	0.989	0.6	0	0.999	1

iii) $K = 33$ jobs:

	MVABLO	Simulation	Std. Dev.	δ_1(%)	NOCAP	δ_2(%)
$\bar{r}_1$	13.254	13.05	0.02	1.5	31.5	141.3
$\bar{r}_2$	15.927	14.49	0.04	9.9	1	93
$\bar{r}_3$	8.455	8.56	0.07	1.2	0.5	94
$\bar{k}_1$	11.621	11.93	0.02	2.5	31.5	164
$\bar{k}_2$	13.965	13.25	0.04	5.3	1	92.4
$\bar{k}_3$	7.413	7.83	0.07	5.3	0.5	93.6
λ	0.8683	0.986	0.2	11.9	1	1.4

Example 2

$N = 3$ stations (serially switched); $M_1 = 4$, $M_2 = 5$, $M_3 = 5$; $1/\mu_1 = 1.5$, $1/\mu_2 = 2.0$, $1/\mu_3 = 1.0$.

i) $K = 10$ jobs:

	MVABLO	Simulation	Std. Dev.	δ_1(%)	NOCAP	δ_2(%)
$\bar{r}_1$	6.925	7.854	0.6	11.4	4.888	37.7
$\bar{r}_2$	8.061	8.206	1.4	1.7	13.650	69.3
$\bar{r}_3$	4.185	5.277	2.4	20.6	1.914	63.7
$\bar{k}_1$	3.683	3.612	0.5	1.9	2.389	35
$\bar{k}_2$	4.204	3.848	1.4	9.2	6.674	73.4
$\bar{k}_3$	2.183	2.47	2.4	11.6	0.936	62.1
λ0.521	0.468	0.5	1.4	11.3	0.488	4.2

ii) $K = 12$ jobs:

	MVABLO	Simulation	Std. Dev.	δ_1(%)	NOCAP	δ_2(%)
$\bar{r}_1$	7.967	8.687	0.33	8.2	5.251	39.5
$\bar{r}_2$	9.019	10.030	0.59	10	17.090	70.4
$\bar{r}_3$	8.011	8.780	0.76	8.7	1.952	77.7
$\bar{k}_1$	3.825	3.792	0.33	0.8	2.593	31.6
$\bar{k}_2$	4.330	4.378	0.60	1	8.442	92.8
$\bar{k}_3$	3.846	3.831	0.70	0.3	0.964	74.8
λ	0.48	0.436	0.3	7.8	0.493	13

Example 3

$N = 3$ stations (Central Server Model); $M_1 = 8$, $M_2 = 7$, $M_3 = 6$; $1/\mu_1 = 0.2$, $1/\mu_2 = 1.2$, $1/\mu_3 = 1.4$; $p_{1j} = 0.5$; $p_{j1} = 1$ for $j = 2, 3$.

i) K = 10 jobs:

	MVABLO	Simulation	Std. Dev.	$\delta_1(\%)$	NOCAP	$\delta_2(\%)$
$\bar{t}_1$	1.674	1.371	2.8	18	0.2740	80
$\bar{t}_2$	5.007	5.199	2.8	3.7	5.0070	3.7
$\bar{t}_3$	7.639	6.885	2.7	10.9	9.0397	31.2
$\bar{k}_1$	2.093	1.850	0.2	13.1	0.3755	79.7
$\bar{k}_2$	3.130	3.580	1.3	12.5	3.4308	4.1
$\bar{k}_3$	4.776	4.593	1.1	3.9	6.1936	34.8
λ	1.25	1.338	2.6	6.5	1.370	2.4

ii) K = 12 jobs:

	MVABLO	Simulation	Std. Dev.	$\delta_1(\%)$	NOCAP	$\delta_2(\%)$
$\bar{t}_1$	2.166	2.435	7.8	11	0.276	88.6
$\bar{t}_2$	5.372	5.781	8.7	7	5.603	3.1
$\bar{t}_3$	6.567	7.265	1.2	9.6	11.131	53.2
$\bar{k}_1$	3.195	3.262	10.9	2	0.383	88.2
$\bar{k}_2$	3.962	3.867	0.8	2.4	3.889	0.5
$\bar{k}_3$	4.843	4.870	6.7	0.5	7.727	58.6
λ	1.474	1.339	8.7	10.1	1.388	3.6

Example 4

$N = 3$ stations; $M_1 = 6$, $M_2 = 8$, $M_3 = 6$; $1/\mu_1 = 2.5$, $1/\mu_2 = 1.2$, $1/\mu_3 = 1.0$; $p_{12} = 0.5$; $p_{21} = 0.7$; $p_{31} = 0.7$; $p_{13} = 0.5$; $p_{23} = 0.3$; $p_{32} = 0.3$.

i) K = 8 jobs:

	MVABLO	Simulation	Std. Dev.	$\delta_1(\%)$	NOCAP	$\delta_2(\%)$
$\bar{t}_1$	13.2517	14.4569	3.1	8.3	17.708	22.4
$\bar{t}_2$	3.5447	4.2267	0.8	16.1	1.821	56.9
$\bar{t}_3$	3.0371	3.6230	1.0	16.1	1.398	61.4
$\bar{k}_1$	5.905	5.7657	0.6	2.4	7.081	22.8
$\bar{k}_2$	1.128	1.2026	0.3	6.2	0.520	56.7
$\bar{k}_3$	0.9667	1.0316	2.8	6.2	0.399	61.2
λ	0.4456	0.3988	2.6	11.7	0.399	0.2

ii) K = 10 jobs:

	MVABLO	Simulation	Std. Dev.	$\delta_1(\%)$	NOCAP	$\delta_2(\%)$
$\bar{t}_1$	14.226	14.6089	6.9	2.6	22.6976	55.3
$\bar{t}_2$	7.032	8.0578	2.6	12.7	1.8254	77.3
$\bar{t}_3$	6.324	6.4840	7.6	2.4	1.3997	78.4
$\bar{k}_1$	5.978	5.8440	0.3	2.3	9.0785	55.3
$\bar{k}_2$	2.113	2.3069	2.3	8.3	0.5215	77.3
$\bar{k}_3$	1.900	1.8485	3.8	2.8	0.3990	78.4
λ	0.4208	0.400	6.6	5.2	0.399	0

iii) K = 11 jobs:

	MVABLO	Simulation	Std. Dev.	$\delta_1(\%)$	NOCAP	$\delta_2(\%)$
$\bar{t}_1$	14.9643	14.6718	8.3	1.9	25.1964	71.7
$\bar{t}_2$	9.9864	9.9777	13.3	0	1.8258	81.7
$\bar{t}_3$	9.1506	7.9690	4.4	14.8	1.3990	82.4
$\bar{k}_1$	5.7488	5.8677	0.3	2	10.0780	71.7
$\bar{k}_2$	2.7403	2.8538	3.9	3.9	0.5216	81.7
$\bar{k}_3$	2.5110	2.2780	4.1	10.2	0.3999	82.4
λ	0.3842	0.3999	8.6	3.9	0.3999	0

Example 5

$N = 3$ stations; $M_1 = 6$, $M_2 = 6$, $M_3 = 6$; $1/\mu_1 = 2.5$, $1/\mu_2 = 1.2$, $1/\mu_3 = 1$; $p_{12} = 0.8$; $p_{21} = 0.7$; $p_{31} = 0.7$; $p_{13} = 0.2$; $p_{23} = 0.3$; $p_{32} = 0.3$

i) K = 8 jobs:

	MVABLO	Simulation	Std. Dev.	$\delta_1(\%)$	NOCAP	$\delta_2(\%)$
$\bar{t}_1$	14.8739	14.0857	0.8	5.5	17.3739	23.3
$\bar{t}_2$	5.4820	4.7029	5.2	16.5	2.1742	53.7
$\bar{t}_3$	2.9308	2.9961	1.1	2.1	1.2385	58.6
$\bar{k}_1$	5.5418	5.6390	1.9	1.7	6.9399	23
$\bar{k}_2$	1.9303	1.7800	5.4	8.4	0.8207	53.8
$\bar{k}_3$	0.5280	0.5809	1.5	9.1	0.2392	58.8
λ	0.3726	0.4003	1.1	6.9	0.3994	0

ii) K = 10 jobs:

	MVABLO	Simulation	Std. Dev.	$\delta_1(\%)$	NOCAP	$\delta_2(\%)$
$\bar{t}_1$	14.2597	14.60780	7	2.3	22.3375	52.9
$\bar{t}_2$	9.3476	9.16352	1.7	2.0	2.1905	76
$\bar{t}_3$	4.1795	4.39325	16.7	4.8	1.2395	71.7
$\bar{k}_1$	5.6778	5.75745	0.8	1.3	8.9325	55.1
$\bar{k}_2$	3.5175	3.40713	4.2	3.2	0.8278	75.7
$\bar{k}_3$	0.8047	0.83542	11.5	3.6	0.2396	71.3
λ	0.3982	0.39412	6.2	1	0.3998	1.4

iii) K = 11 jobs:

	MVABLO	Simulation	Std. Dev.	$\delta_1(\%)$	NOCAP	$\delta_2(\%)$
$\bar{t}_1$	14.1946	14.8849	4.8	4.6	24.8312	66.8
$\bar{t}_2$	10.3825	11.6514	2.7	10.8	2.1934	81.1
$\bar{t}_3$	4.3431	4.6866	3.5	7.3	1.2396	73.5
$\bar{k}_1$	5.9809	5.8099	0.8	2.9	9.9312	70.9
$\bar{k}_2$	4.1343	4.3046	2.6	3.9	0.8290	80.7
$\bar{k}_3$	.8848	0.8854	7.5	0	0.2397	73
λ	0.4213	0.3913	5.6	7.9	0.3999	2.4

Example 6

$N = 4$ Stations, (serially switched); $M_1 = 4$, $M_2 = 4$; $M_3 = 4$; $M_4 = 4$; $1/\mu_1 = 1.8$, $1/\mu_2 = 2.6$, $1/\mu_3 = 2.8$, $1/\mu_4 = 2.4$.

i) K = 10 jobs:

	MVABLO	Simulation	Std. Dev.	$\delta_1(\%)$	NOCAP	$\delta_2(\%)$
$\bar{t}_1$	3.792	5.087	2.1	25	3.792	25
$\bar{t}_2$	9.339	10.660	1.9	12	9.339	12
$\bar{t}_3$	11.784	10.840	1.6	9	11.784	9
$\bar{t}_4$	7.426	9.530	3.7	25	7.426	25
$\bar{k}_1$	1.172	1.404	1.3	16	1.172	16
$\bar{k}_2$	2.886	2.952	1.7	3	2.886	3
$\bar{k}_3$	3.644	3.003	2.4	22	3.644	22
$\bar{k}_4$	2.296	2.641	2.1	13	2.296	13
λ	0.309	0.276	2.1	3.3	0.309	3.3

ii) K = 12 jobs:

	MVABLO	Simulation	Std. Dev.	$\delta_1(\%)$	NOCAP	$\delta_2(\%)$
$\bar{t}_1$	8.0907	10.600	1.3	23.6	4.016	62
$\bar{t}_2$	11.236	12.490	0.5	10	10.873	13
$\bar{t}_3$	8.997	10.12	1.4	11	14.266	41
$\bar{t}_4$	8.379	8.643	1.7	3	8.379	3
$\bar{k}_1$	2.645	3.039	1.3	12.9	1.284	58
$\bar{k}_2$	3.673	3.583	0.5	2.5	3.476	2.9
$\bar{k}_3$	2.941	2.902	1.4	1.3	4,.561	57.1
$\bar{k}_4$	2.739	2.477	1.7	10.5	2.678	8.1
λ	0.326	0.286	1.3	13.9	0.319	11.5

iii) $K = 14$ jobs:

	MVABLO	Simulation	Std. Dev.	$\delta_1(\%)$	NOCAP	$\delta_2(\%)$
$\bar{t}_1$	14.459	13.57	1.1	6.5	4.196	69
$\bar{t}_2$	12.902	14.42	2.2	10.5	12.397	14
$\bar{t}_3$	9.655	13.11	4.7	26.3	16.916	29
$\bar{t}_4$	13.976	12.59	3.2	11	9.261	26.4
$\bar{k}_1$	3.969	3.538	1.3	12.2	1.373	61.1
$\bar{k}_2$	3.542	3.76	1.1	5.7	4.057	7.9
$\bar{k}_3$	2.651	3.418	2.1	22.4	5.537	62
$\bar{k}_4$	3.837	3.285	3.7	16.8	3.031	7.7
λ	0.274	0.26	1.1	5.5	0.327	25.7

Example 7

$N = 4$ Stations, (serially switched); $M_1 = 3$; $M_2 = 6$; $M_3 = 7$; $M_4 = 5$; $1/\mu_1 = 1.8$, $1/\mu_2 = 2.6$, $1/\mu_3 = 2.8$, $1/\mu_4 = 2.4$;

i) $K = 15$ jobs:

	MVABLO	Simulation	Std. Dev.	$\delta_1(\%)$	NOCAP	$\delta_2(\%)$
$\bar{t}_1$	4.273	5.393	3.2	20	4.273	20
$\bar{t}_2$	13.150	15.640	1.8	16	13.150	16
$\bar{t}_3$	18.310	19.130	0.7	4	18.310	4
$\bar{t}_4$	9.671	10.890	1.3	11	9.671	11
$\bar{k}_1$	1.411	1.582	2.4	10	1.411	10
$\bar{k}_2$	4.344	4.595	1.2	5	4.344	5
$\bar{k}_3$	6.049	5.625	3.1	7	6.049	7
$\bar{k}_4$	3.195	3.197	0.7	0	3.195	0
λ	0.33	0.293	3.2	12.6	0.33	12.6

ii) $K = 18$ jobs:

	MVABLO	Simulation	Std. Dev.	$\delta_1(\%)$	NOCAP	$\delta_2(\%)$
$\bar{t}_1$	7.857	8.484	1.1	7.3	4.457	47.4
$\bar{t}_2$	18.002	20.29	0.3	11.2	15.360	24.3
$\bar{t}_3$	17.280	22.09	0.5	21.7	22.709	2.8
$\bar{t}_4$	12.612	14.02	0.8	10	10.812	22.8
$\bar{k}_1$	2.5369	2.354	1.1	7.7	1.504	36
$\bar{k}_2$	5.812	5.629	0.3	3.2	5.183	7.9
$\bar{k}_3$	5.579	6.13	0.5	8.9	7.663	25
$\bar{k}_4$	4.072	3.88	0.8	4.4	3.648	10.3
λ	0.322	0.276	1.1	16.7	0.337	22

iii) $K = 18$ jobs:

	MVABLO	Simulation	Std. Dev.	$\delta_1(\%)$	NOCAP	$\delta_2(\%)$
$\bar{t}_1$	12.933	11.67	1.3	10.8	4.552	61
$\bar{t}_2$	27.597	24.81	2.2	11.2	16.782	32
$\bar{t}_3$	31.095	28.38	1.3	9.5	25.845	8.9
$\bar{t}_4$	20.893	19.69	1.1	6.1	11.484	41.6
$\bar{k}_1$	2.795	2.762	0.3	1.2	1.5521	43.8
$\bar{k}_2$	5.965	5.869	0.9	1.6	5.721	2.5
$\bar{k}_3$	6.722	6.715	0.8	0	8.811	31.2
$\bar{k}_4$	4.516	4.657	2.1	3	3.915	16
λ	0.216	0.236	2.2	8.4	0.34	44

Example 8

$N = 5$ Stations; (serially switched); $M_1 = 2$; $M_2 = 4$; $M_3 = 3$; $M_4 = 4$; $M_5 = 2$; $1/\mu_1 = 1$; $1/\mu_2 = 2$; $1/\mu_3 = 1.5$; $1/\mu_4 = 1.8$; $1/\mu_5 = 1.6$.

i) $K = 10$ jobs:

	MVABLO	Simulation	Std. Dev.	$\delta_1(\%)$	NOCAP	$\delta_2(\%)$
$\bar{t}_1$	1.6853	2.512	2.9	32.9	1.6853	32.9
$\bar{t}_2$	7.9283	7.395	1.4	7.2	7.9283	7.2
$\bar{t}_3$	3.6759	4.761	2.1	22.7	3.6759	22.7
$\bar{t}_4$	5.8070	7.430	1.5	21.8	5.8070	21.8
$\bar{t}_5$	4.2781	3.323	1.5	28.7	4.2781	28.7
$\bar{k}_1$	0.7210	0.988	2.9	27	0.7210	27
$\bar{k}_2$	3.3918	2.910	1.4	16.5	3.3918	16.5
$\bar{k}_3$	1.5726	1.873	2.1	16	1.5726	16
$\bar{k}_4$	2.4843	2.921	1.5	14.9	2.4843	14.9
$\bar{k}_5$	1.8302	1.307	1.6	40	1.8302	40
λ	0.427	0.393	2.9	8.8	0.427	8.8

ii) $K = 12$ jobs:

	MVABLO	Simulation	Std. Dev.	$\delta_1(\%)$	NOCAP	$\delta_2(\%)$
$\bar{t}_1$	3.7523	3.460	1.3	8.4	1.7523	49.3
$\bar{t}_2$	7.6793	9.696	3.6	20.7	9.6793	0
$\bar{t}_3$	4.0303	5.556	2.1	27.4	4.0303	27.4
$\bar{t}_4$	8.3349	9.320	3.5	10.5	6.7349	27.7
$\bar{t}_5$	3.1672	3.862	1.4	17.9	4.7670	23.4
$\bar{k}_1$	1.6699	1.302	1.3	28.2	0.7798	40
$\bar{k}_2$	3.4176	3.649	3.6	6.3	4.3076	18
$\bar{k}_3$	1.7936	2.090	2.1	14.1	1.7936	14.1
$\bar{k}_4$	3.7093	3.507	3.5	5.4	2.9972	14.5
$\bar{k}_5$	1.4095	1.452	1.4	2.9	2.1215	46
λ	0.445	0.375	3.8	18.6	0.445	18.6

iii) $K = 14$ jobs:

	MVABLO	Simulation	Std. Dev.	$\delta_1(\%)$	NOCAP	$\delta_2(\%)$
$\bar{t}_1$	5.271	5.413	0.77	2.6	1.804	66.6
$\bar{t}_2$	10.018	11.830	0.16	15.3	11.591	2
$\bar{t}_3$	7.286	8.068	0.59	9.6	4.339	46.2
$\bar{t}_4$	10.111	11.540	0.29	12.3	7.651	33.7
$\bar{t}_5$	5.448	5.246	0.82	3.8	5.210	0
$\bar{k}_1$	1.935	1.800	0.77	7.5	0.825	54.1
$\bar{k}_2$	3.677	3.935	0.16	6.5	5.303	34.7
$\bar{k}_3$	2.674	2.682	0.61	0.2	1.985	25.9
$\bar{k}_4$	3.712	3.839	0.32	3.3	3.501	8.8
$\bar{k}_5$	2	1.744	0.81	14.6	2.384	36.6
λ	0.367	0.332	0.9	10.5	0.457	37.6

Example 9

$N = 3$ Stations, (serially switched); $M_1 = 35$; $M_2 = 30$; $M_3 = 40$; $1/\mu_1 = 10$, $1/\mu_2 = 100$, $1/\mu_3 = 50$.

i) $K = 50$ jobs:

	MVABLO	Simulation	Std. Dev.	$\delta_1(\%)$	NOCAP	$\delta_2(\%)$
$\bar{t}_1$	1895.23	1756.00	1.3	7.9	11.1	99.3
$\bar{t}_2$	2948.00	2820.00	1.2	4.5	4888.8	73.3
$\bar{t}_3$	99.60	120.00	2.5	16.9	100.0	16.7
$\bar{k}_1$	19.17	18.75	1.1	2.2	0.1	99.4
$\bar{k}_2$	29.82	29.98	0.6	0	48.8	63.1
$\bar{k}_3$	1.01	1.26	2.2	20.3	1.0	21
λ	0.01	0.01	0.1	0	0.01	0

ii) $K = 75$ jobs:

	MVABLO	Simulation	Std. Dev.	$\delta_1(\%)$	NOCAP	$\delta_2(\%)$
$\bar{t}_1$	3549.64	3198.00	2.4	10.9	11.11	99.6
$\bar{t}_2$	2601.73	2821.00	1.6	7.7	7388.88	162
$\bar{t}_3$	1460.92	1010.00	7.9	44.5	100.00	90
$\bar{k}_1$	34.97	34.23	1.1	2.1	0.11	99.6
$\bar{k}_2$	25.63	29.99	7.5	33.6	73.88	146
$\bar{k}_3$	14.39	10.77	3.4	33.6	1	91
λ	0.01	0.01	0.01	0	0.01	0

iii) K = 100 jobs:

	MVABLO	Simulation	Std. Dev.	δ_1(%)	NOCAP	δ_2(%)
$\bar{t}_1$	3547.600	3234.00	0.2	9.6	11.11	99
$\bar{t}_2$	2577.410	2823.00	0.1	8.6	9888.88	250
$\bar{t}_3$	4078.130	3328.00	0.2	22.5	100.00	97
$\bar{k}_1$	34.769	34.50	0.25	0	0.11	99
$\bar{k}_2$	25.261	29.98	0.1	15.7	1	230
$\bar{k}_3$	39.969	35.52	0.2	12.5	1	97
λ	0.009	0.01	0.2	8	0.01	6

Example 10

$N = 4$ Stations; (serially switched); $M_1 = 20$; $M_2 = 30$; $M_3 = 40$; $M_4 = 15$; $1/\mu_1 = 10$; $1/\mu_2 = 5$; $1/\mu_3 = 20$; $1/\mu_4 = 15$;

i) K = 50 jobs:

	MVABLO	Simulation	Std. Dev.	δ_1(%)	NOCAP	δ_2(%)
$\bar{t}_1$	20.035	20.13	7.2	0	19.990	0
$\bar{t}_2$	131.570	125.90	3.8	4.5	6.667	94.7
$\bar{t}_3$	793.460	785.80	0.2	0	913.334	16.2
$\bar{t}_4$	60.136	56.85	9.6	5.7	59.990	5.5
$\bar{k}_1$	0.996	1.01	7.2	1.7	0.999	1.4
$\bar{k}_2$	6.544	6.38	3.8	2.5	0.333	94.7
$\bar{k}_3$	39.467	39.75	3.8	0	45.660	14.8
$\bar{k}_4$	2.991	2.86	9.6	4.4	2.999	4.7
λ	0.049	0.05	0.1	1.3	0.050	0.7

ii) K = 75 jobs:

	MVABLO	Simulation	Std. Dev.	δ_1(%)	NOCAP	δ_2(%)
$\bar{t}_1$	72.28	66.50	3.8	8.7	20.00	70
$\bar{t}_2$	609.69	569.40	0.6	7.1	6.66	99
$\bar{t}_3$	787.36	789.40	0	0.2	1413.33	79
$\bar{t}_4$	59.45	56.85	9.6	4.5	59.99	6
$\bar{k}_1$	3.54	3.36	3.8	5.5	1.00	70
$\bar{k}_2$	29.91	28.85	0.6	3.6	0.33	99
$\bar{k}_3$	38.62	39.93	0	3.2	70.66	77
$\bar{k}_4$	2.92	2.86	9.6	1.8	3.00	5
λ	0.05	0.05	0.7	0	0.05	0

iii) K = 100 jobs:

	MVABLO	Simulation	Std. Dev.	δ_1(%)	NOCAP	δ_2(%)
$\bar{t}_1$	405.68	389.90	0.5	4.0	20.00	95
$\bar{t}_2$	705.54	612.40	0.1	15.2	6.67	99
$\bar{t}_3$	931.45	824.70	0.1	12.9	1913.33	132
$\bar{t}_4$	312.50	239.60	0.8	30.4	60.00	75
$\bar{k}_1$	17.22	18.85	0.5	8.6	1.00	95
$\bar{k}_2$	29.95	29.63	0.1	1.1	0.33	99
$\bar{k}_3$	39.55	39.94	0.5	0.9	95.66	139
$\bar{k}_4$	13.26	11.58	0.8	14.5	3.00	74
λ	0.04	0.05	0.5	12.0	0.05	3

References

[1] I. F. Akyildiz, "Exact product form solution for queueing networks with blocking," *IEEE Trans. Comput.*, vol. C-36, pp. 122–125, Jan. 1987.

[2] —, "On the exact and approximate throughput analysis of closed queueing networks with blocking," *IEEE Trans. Software Eng.*, vol. 14, pp. 62–71, Jan. 1988.

[3] —, "Product form approximations for queueing networks with multiple servers and blocking," *IEEE Trans. Comput.*, to be published.

[4] I. F. Akyildiz and S. Kundu, "Buffer allocation in deadlock free blocking queueing networks," Dep. Comput. Sci., Louisiana State Univ., Tech. Rep. TR-86-013, Apr. 1986.

[5] T. Altiok, "Approximate analysis of exponential tandem queues with blocking," *European J. Oper. Res.*, vol. 11, pp. 390–398, Oct. 1982.

[6] T. Altiok and H. G. Perros, "Approximate analysis of arbitrary configurations of open queueing networks with blocking," *AIIE Trans.*, Mar. 1986.

[7] Y. Bard, "Some extension to multiclass queueing network analysis," in *Fourth Int. Conf. Modeling and Performance Evaluation of Computer Systems*, vol. 1, Vienna, Austria, Feb. 1979.

[8] F. Baskett, K. M. Chandy, R. R. Muntz, and G. Palacios, "Open, closed and mixed network of queues with different classes of customers," *J. ACM*, vol. 22, no. 2, pp. 248–260, Apr. 1975.

[9] A. Brandwajn and Y. L. Jow, "Tandem exponential queues with finite buffers," in *Proc. Computer Networking and Performance Evaluation*, T. Hasegawa, H. Takagi, and Y. Takahashi, Eds. Amsterdam, The Netherlands: North-Holland, 1985, pp. 245–258.

[10] L. Gün and A. Mckowski, "Matrix-geometric solution for finite capacity queues with phase-type distributions," in *Proc. Performance 87 Conf.*, P. J. Courtois and G. Latouche, Eds. Amsterdam, The Netherlands: North-Holland, 1988, pp. 269–282.

[11] R. O. Onvural and H. G. Perros, "Some exact results on closed queueing networks with blocking," Dep. Comput. Sci., NCSU, Tech. Rep., 1987.

[12] R. O. Onvural, "A survey of closed queueing networks with finite buffers," Dep. Comput. Sci., NCSU, Tech. Rep., 1987.

[13] H. G. Perros, "Queueing networks with blocking: A bibliography," *ACM Sigmetrics Perform. Eval. Rev.*, Aug. 1984.

[14] H. G. Perros and T. Altiok, "Approximate analysis of open networks of queues with blocking: tandem configurations," *IEEE Trans. Software Eng.*, vol. SE-12, pp. 450–462, Mar. 1986.

[15] H. G. Perros, A. Nilsson, and Y. Liu, "Approximate analysis of product form type queueing networks with blocking and deadlock," *Perform. Eval.*, to be published.

[16] M. Reiser and S. S. Lavenberg, "Mean value analysis of closed multichain queueing networks," *J. ACM*, vol. 27, no. 2, pp. 313–322, Apr. 1980.

[17] C. H. Sauer and K. M. Chandy, *Computer Systems Performance Modeling*. Englewood Cliffs, NJ: Prentice-Hall, 1981.

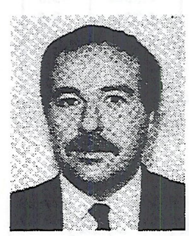
[18] P. J. Schweitzer and T. Altiok, "Aggregate modeling of tandem queues with blocking," in *Proc. Second Int. Workshop Applied Math. and Performance/Reliability Models of Computer/Communication Systems*, Univ. Rome, Italy, May 1987.

[19] K. C. Sevcik and I. Mitrani, "The distribution of queueing network states at input and output instants," *J. ACM*, vol. 28, no. 2, pp. 358–371, Apr. 1981.

[20] W. Stewart, "A comparison of numerical techniques in Markov modeling," *Commun. ACM*, vol. 21, no. 2, pp. 144–152, Feb. 1978.

[21] R. Suri and G. W. Diehl, "A variable buffer-size model and its use in analyzing closed queueing networks with blocking," *Management Sci.*, vol. 32, pp. 206–225, Feb. 1986.

[22] Y. Takahashi, H. Miyahara, and T. Hasegawa, "An approximation method for open restricted queueing networks," *Oper. Res.*, vol. 28, no. 3, pp. 594–602, May-June 1980.



Ian F. Akyildiz (M'86) received the Diplom Informatiker and Doctor of Engineering degrees in computer science from the University of Erlangen-Nuernberg, West Germany, in 1981 and 1984, respectively.

From 1981 through 1985 he served as an Assistant Professor (Wissenschaftlicher Angestellter) in the Department of Operating Systems (Informatik IV) at the University of Erlangen-Nuernberg. During that time he coauthored a textbook entitled *Analysis of Computer Systems* (in German) published by Teubner-Verlag in the Fall of 1982. He also served as a consultant and lecturer for the University of Maryland European Division and Grundig Academy in West Germany from 1983 through 1985. In January of 1985 he joined the faculty of the Department of Computer Science at Louisiana State University, Baton Rouge, as an Assistant Professor. He was also a Visiting Professor in the Department of Computer Science at the University of Florida in the Summer of 1985 and in the Department of Computer Science Department of the Universidad Tecnica de Federico Santa Maria in Valparaiso, Chile, in the Summer of 1986. In the Fall of 1987 he joined the faculty of the School of Information and Computer Science at Georgia Institute of Technology as an Assistant Professor. He is preparing a textbook entitled *Computer Systems Performance Analysis* published by Wiley, Chichester, England. His research interests are performance evaluation, operating systems, distributed systems, and computer networks.

Dr. Akyildiz is a member of the Association for Computing Machinery (Sigops and Sigmetrics), Gesellschaft fuer Informatik (GI), and the German Interest Group in Measurement, Modeling, and Evaluation of Computer Systems (MMB).

ation and Consulting Academy in West Germany from 1981 through 1985. In January of 1985 he joined the faculty of the Department of Computer Science at Louisiana State University, Baton Rouge, as an Assistant Professor. He was also a Visiting Professor in the Department of Computer Science at the University of Florida in the Summer of 1985 and in the Department of Computer Science Department of the Universidad Tecnica de Federico Santa Maria in Valparaiso, Chile, in the Summer of 1988. In the Fall of 1987 he joined the faculty of the School of Information and Computer Science at Georgia Institute of Technology as an Assistant Professor. He is preparing a textbook entitled Computer System Performance Analysis published by Wiley, Chichester, England. His research interests are performance evaluation, operating systems, distributed systems, and computer networks.

Dr. Akyildiz is a member of the Association for Computing Machinery (Sigmis and Sigmetrics), Gesellschaft fuer Informatik (GI), and the German Interest Group in Measurement, Modelling, and Evaluation of Computer Systems (MMB).