Proceedings of the 2007 Industrial Engineering Research Conference
G. Bayraksan, W. Lin, Y. Son, and R. Wysk, eds.

Approximate Mean Value Analysis for Closed Queuing Networks with Multiple-Server Stations

Rajan Suri, Sushanta Sahu
Department of Industrial and Systems Engineering
University of Wisconsin-Madison, Wisconsin 53706, USA

Mary Vernon
Department of Computer Sciences
University of Wisconsin-Madison, Wisconsin 53706, USA

Abstract

Closed Queueing Networks are used in modeling various systems such as FMS, CONWIP Material Control, Computer/Communication Systems, and Health Care. Mean Value Analysis (MVA) is often used to compute the performance measures for these models. For networks with multiple-server stations, the exact MVA algorithm becomes computationally complex and existing approximations introduce high errors. The Schweitzer-Bard (S-B) approximation for MVA is simple and computationally efficient. However it has only been developed for networks with single-server stations. We provide an extension to S-B MVA to enable the analysis of networks with multiple-server stations. Comparison with simulation demonstrates the accuracy of our approach.

Keywords: Closed Queuing Networks, Multiple-Server Stations, Mean Value Analysis, Approximations

1. Introduction and Motivation

A closed queuing network (CQN) can be used to represent many systems. Examples include Flexible Manufacturing Systems (FMS), Biotech Manufacturing Systems, CONWIP Material Control, Computer/Communication Systems, and Health Care Systems. Exact analysis of CQNs is possible for networks that satisfy the product-form structure [1]. For these networks, computation of steady state performance measures such as server utilization and mean queue lengths requires the computation of a normalization constant. However, due to the large number of states even for moderately sized networks, obtaining the normalization constant requires a great deal of computational effort. Efficient computational algorithms such as convolution [2] and mean value analysis (MVA) [3] were developed to overcome this problem. With convolution, however, numerical difficulties arise when the network has a large number of stations and customer classes, and the method also does not lend itself to heuristic extensions for more general system models. Mean value analysis, which avoids these problems, is therefore often used to compute the performance measures. However, the storage requirements for MVA increase for networks with many service stations, customer classes, and customers within each class. Furthermore, if the network contains multiple-server stations, the solution requires the evaluation of marginal probabilities in addition to the mean values, thereby making the solution more complex and increasing the storage requirements further.

Some approximations have been developed to reduce the computational complexity of MVA. For networks with multiple-server stations an approximation was proposed by Chandy and Neuse [5], and improved by Akyildiz and Bolch [6]. A key feature of these approximations is that instead of determining the probability mass for the queue length distributions exactly, it is placed as close as possible to the mean queue length estimate. In addition to introducing high errors, this method is still complicated enough to have limited use in practical applications. For networks with single-server stations, the Schweitzer-Bard (S-B) approximation (also known as approximate MVA or AMVA) improves the computational efficiency of the MVA algorithm [7] [8]. The basic assumption (for single class networks) in this approximation is that when a customer is removed from a CQN, the proportion in which the customers are distributed across the network does not change. AMVA is very simple; however, it has only been developed for networks with single-server stations.

We attempt to develop a simple approximate method, which is an extension to AMVA, for enabling the analysis of networks with multiple-server stations. On the one hand the method retains the simplicity of the AMVA algorithm and on the other it strives for accuracy in performance predictions. We compare our method with simulation for a variety of networks to demonstrate the accuracy of our approach. Note that throughout this paper we will assume exponential service time distribution and FCFS scheduling discipline.

2. Overview of Strategy for Developing New Approximation

The notation used in the development of this method is given in Table 1:

Table 1: Notation for model development

Given Parameters	**Steady State Performance Measures**
K - Number of stations in the network	$R_k(N)$ - Mean response time per visit to station k
N - Customer population in the network	$Q_k(N)$ - Mean queue length (including customers in service) at station k
C_k - Number of servers at station k	
V_k - Mean number of visits by a customer to station k	$X_k(N)$ - Throughput of station k
	$X(N)$ - Throughput of network
T_k - Mean service time per visit to station k	$\rho_k(N)$ - Utilization of each server at station k

MVA is based on the arrival theorem [4] which states that the queue length at station k, as seen by an arriving customer, is given by $Q_k(N-1)$. Thus, the response time for this customer at a single-server station is given by [3]

$$R_k(N) = T_k + T_k Q_k(N-1) \tag{1}$$

If station k has multiple servers then this arriving customer may find some customers in service and some waiting. Customers at the multiple servers in the station would result in multiple residual service times. Also, the waiting time in the queue will now be less than the waiting time if there were only one server. As a way of dealing with these complications, and to capture the reduction in waiting times at the multiple-server station, we propose to use a correction factor, denoted by Y_k, in conjunction with the AMVA algorithm. We propose the following equation for the mean response time for an arriving customer, per visit to a multi-server station k:

$$R_k(N) = T_k + Y_k T_k Q_k(N-1) \tag{2}$$

We give the following interpretation to equation (2): When a customer circulating in the network arrives at a multi-server station k, the mean response time is equal to its own service time, T_k, added to its mean waiting time. The mean waiting time is the product of the mean number of customers that the arriving customer sees in the station and a reduced interference time, $Y_k T_k$, for each of those customers, to capture the effect of multiple servers in the station. Using the S-B approximation, equation (2) becomes

$$R_k(N) = T_k + Y_k T_k \frac{N-1}{N} Q_k(N) \tag{3}$$

The throughput of the network and the mean queue length at station k are given by

$$X(N) = N \Big/ \sum_{k=1}^{K} V_k R_k(N) \tag{4}$$

$$Q_k(N) = V_k X(N) R_k(N) \tag{5}$$

On rewriting equation (5) and substituting for $X(N)$ from equation (4) we obtain

$$Q_k(N) = R_k(N) V_k N \Big/ \sum_{k=1}^{K} V_k R_k(N) \tag{6}$$

If we know Y_k, on substituting equation (3) into equation (6) we get a system of k non-linear equations in the k unknowns $Q_k(N)$. The values of $Q_k(N)$ for the stations are calculated by solving these equations, which ultimately allows for the calculation of the network throughput. In order that equations (3) to (5) can be solved to obtain the values of the performance measures, we need to have an expression for Y_k. In keeping with the product-form solution structure, we would like Y_k to be a function of the local station parameters. From elementary queuing theory, the waiting time should be influenced by the number of servers C_k at station k and the station utilization ρ_k. Therefore, in an effort to derive the simplest possible approximation we hypothesize that Y_k can be expressed as a function of only these two parameters, i.e.

$$Y_k = f(C_k, \rho_k) \tag{7}$$

To determine the form of Y_k, we first analyze a simple 2-station balanced network under various conditions and obtain a candidate function. Then we test an algorithm based on this function for a variety of complex networks.

3. Analysis of a Balanced 2-Station Network

We consider a 2-station network in which the first station has one server ($C_1 = 1$) and the second station has multiple servers (C_2). To obtain a balanced network, let V_1=V_2=1, and let $T_1 = 1$, $T_2 = C_2$. Henceforth, for simplicity of notation, we refer to C_2 as C and Y_2 as Y. Also (since the network is balanced) denote the utilization of both stations by ρ. On substituting equation (5) into (3), substituting C_2 for T_2 and rearranging the terms, we express Y as

$$Y = \frac{R_2(N) - C}{\frac{N-1}{N} C X(N) R_2(N)} \tag{8}$$

The station utilization is related to the network throughput as follows

$$\rho(N) = X(N) T_2 / C \tag{9}$$

Using this and the fact that $T_2 = C$, equation (8) becomes

$$Y = \frac{R_2(N) - C}{\frac{N-1}{N} C \rho(N) R_2(N)} \tag{10}$$

Since the exact values of $R_2(N)$ and $\rho(N)$ can be calculated analytically via convolution [2], this expression gives us the value of Y which would result in the exact throughput for a given set of parameters. Next we attempt to ascertain whether Y can be expressed simply in terms of C and ρ. To do this, for the 2-station network as described previously, we obtain $R_2(N)$ and $\rho(N)$ analytically for values of C from 2 to 11 and N from $C+1$ to $C+15$ for each C. We next calculate value of Y for each of these settings using equation (10), and plot Y as a function of ρ for each C. From the shapes of the plots, and using insights from Sakasegawa's approximation [9], we hypothesize a function of the form $Y = a\rho^b$. However, since we have different curves for each C, the parameters a and b should depend on C. Hence the complete functional form is hypothesized to be

$$Y(C, \rho) = a(C) \rho^{b(C)} \tag{11}$$

Next we verify if such a function can fit the data reasonably well. First, we find a form for the function $a(C)$ using a boundary condition. Then, for each value of C we find the values of $b(C)$ that fit the data best. This gives us a set of values $b(2), b(3), ..., b(11)$, which we use to find a form for the function $b(C)$.

3.1 Determining Functional Form for $a(C)$

On dividing the numerator and denominator of equation (10) by R_2 and on rearranging the terms, we get

$$Y = \frac{1 - (C/R_2)}{C[1 - (1/N)]\rho} \tag{12}$$

Now as $N \to \infty$, $\rho \to 1$ and $R_2 \to \infty$. Using these limits in equation (12), we have

$$\lim_{\rho \to 1} Y = 1/C \tag{13}$$

On combining equations (11) and (13) we see that

$$a(C) = 1/C \tag{14}$$

3.2 Determining Functional Form for $b(C)$

We first estimate the numerical values of $b(C)$. We use equation (14) to express the correction factor Y as

$$Y = \frac{1}{C}\rho^{b(C)} \tag{15}$$

For each value of C, we first use the observed data $(\rho$ and $Y)$ for a linear regression on the logarithmic form of equation (15) (i.e., $\log(CY) = b(C)\log\rho$) and get the best fit values for $b(C)$. Next, we plot the $b(C)$ values against the corresponding C values. Note that $b(C) = 0$ at $C = 1$. To prove this, we show that for a balanced network, $Y = 1$ at $C = 1$ as follows: Based on the arrival theorem, for the balanced 2-station network with a single server at each station, an arriving customer should see $(N-1)/2$ customers on average at each station. Since there is a single server in station 2, from the second term of the right hand side of equation (3),

$$Y \frac{N-1}{N} Q_2(N) = \frac{N-1}{2} \tag{16}$$

Since the network is balanced $Q_2(N) = N/2$ and putting this in equation (16) we see that Y equals 1. Based on the plot of $b(C)$ vs. C and the fact that $Y = 1$ at $C = 1$ (which results in $b(1) = 0$), we guess $b(C)$ to be of the form $b(C) = \alpha(C^{\beta} - 1)$. We use the least squares estimation method using the $b(C)$ data obtained earlier to get $\alpha = 4.464$ and $\beta = 0.676$. $b(C)$ can now be expressed as $b(C) = 4.464(C^{0.676} - 1)$. A comparison of this expression with actual $b(C)$ data shows that the curve fits the data well. Putting together the expressions for $a(C)$ and $b(C)$, we obtain the final form for our correction factor as

$$Y = \frac{1}{C}\rho^{4.464(C^{0.676}-1)} \tag{17}$$

4. Testing the New Approximation for a Variety of Networks

The solution method using the correction factor is shown next. We call this new method MS-AMVA (Multiple-Server AMVA). A set of non-linear simultaneous equations are solved to obtain the performance measures.

$$Y_k = \frac{1}{C_k}\rho_k^{4.464(C_k^{0.676}-1)}$$

$$R_k(N) = T_k, \quad N \le C_k$$

$$R_k(N) = T_k + Y_k T_k \frac{N-1}{N} Q_k(N), \quad N > C_k$$

$$X(N) = N \Big/ \sum_{k=1}^{K} V_k R_k(N), \quad Q_k(N) = V_k X(N) R_k(N), \quad \rho_k(N) = V_k X(N) T_k / C_k$$

For validation, we first conduct experiments for a balanced 2-station network with one single and one multiple-server station. We assume V_k=1 $\forall k$. We vary C_2 from 2 to 6 in increments of 2 and we also vary the population in increments. Results show that the throughput errors are within 2.5% for all the test cases. We then test the MS-AMVA method for an unbalanced 2-station network. To unbalance the network, we vary T_1 below and above 1 while we keep $T_2 = C_2$ We use four different values of C_2: 2, 4, 6, and 8. N is varied from 6 to 12 in increments of 2. Even when the multiple-server station is highly stressed (has a relatively high utilization), the throughput errors are within 8% of exact values. Next, we test an unbalanced 4-station network with 4 servers in station 4 and one server in each of the other stations. The throughput errors are within 3% for all cases (including high stress cases for the multiple-server station) for the range of parameters selected. From these results it appears that the form of Y that was originally developed for a 2-station balanced network also gives low error values when used to predict throughput of unbalanced and extended networks. We next test MS-AMVA for a 3-station unbalanced network containing one single server and two multi-server stations with $C_1 = 1$, $C_2 = 2$ and $C_3 = 4$. As an example, Table 2 shows the results for $N = 8$. S_i represent the service stations and the service times at these are included in the table. The results show that for the range of parameters selected, errors are within 6.5%.

Table 2: Results for an unbalanced 3-station network with 2 multi-server stations, $N = 8$

Service Time			%Utilization			Throughput		% Error
S_1	S_2	S_3	S_1	S_2	S_3	Exact	MS-AMVA	
1	2	2	84.3	84.3	42.2	0.8434	0.8407	-0.3
1	2	4	74.6	74.6	74.6	0.7463	0.7643	2.4
1	2	8	48.2	48.2	96.5	0.4824	0.4558	-5.5
1	1	2	97.5	48.7	48.7	0.9749	0.9347	-4.1
1	4	2	49.6	99.2	24.8	0.4959	0.4704	-5.1
1	1	4	83.2	41.6	83.2	0.8322	0.8275	-0.6
1	4	4	48.6	97.2	48.6	0.4862	0.4622	-4.9
1	1	8	49.3	24.6	98.5	0.4927	0.4605	-6.5
1	4	8	41.4	82.8	82.8	0.4138	0.4187	1.2

5. Determining the Correction Factor for Multi-Class Networks

We use the following notation:
$\vec{N}$ - Customer population vector (N_m is number of customers of class m)
$T_{k,m}$ - Mean service time for class m per visit to station k
$R_{k,m}(\vec{N})$ - Mean response time for a class m customer per visit to station k with $\vec{N}$ customers in network
$Q_{k,m}(\vec{N})$ - Average queue length for class m customers at station k with $\vec{N}$ customers in network
$X_m(\vec{N})$ - Throughput of class m customers

When there are multiple customer classes, the waiting time of a class m customer at station k is also dependent on the queue length of class r $(r \neq m)$ customers in front of it. If station k has only one server then using the S-B approximation [7] [8] the response time for an arriving class m customer is given by

$$R_{k,m}(\vec{N}) = T_{k,m} + T_{k,m}\frac{N_m - 1}{N_m} Q_{k,m}(\vec{N}) + \sum_{i \neq m} T_{k,i} Q_{k,i}(\vec{N}) \tag{18}$$

If station k has multiple servers, we denote the correction factor to be applied for estimating the waiting times due to the class r customers, by $Y_{k,r}$. We note that $Y_{k,r}$ may or may not be equal to Y_k and therefore needs to be investigated. We first conduct a thought experiment wherein we consider two simple 2-station networks, Network 1 with a single customer class and Network 2 with two customer classes. We set the population, mean service times, and visit counts for the networks in such a way that the networks are equivalent in terms of the response times for the customers. Then we compare the equations for the two networks to express $Y_{k,r}$ in terms of Y_k. We find that the correction factor for a network containing multiple classes with same mean service times has the same form as in the case of a single class network or in other words $Y_{k,r} = Y_k$. We now hypothesize that in a network containing multiple customer classes with different mean service times, the same form for the correction factor will hold. In other words, the mean response time for a class m customer at a multiple-server station k can be approximated by

$$R_{k,m}(\vec{N}) = T_{k,m} + Y_k T_{k,m}\frac{N_m - 1}{N_m} Q_{k,m}(\vec{N}) + \sum_{i \neq m} Y_k T_{k,i} Q_{k,i}(\vec{N}) \tag{19}$$

We also hypothesize that Y_k is a function of the total utilization at station k by all the customer classes. We perform experiments to test our hypotheses and compare our results with simulation, since analytical results do not exist when the network has FCFS service discipline with different mean service times for different customer classes.

6. Validation Experiments for Multi-Class Networks

The solution method using the correction factor uses a set of non-linear simultaneous equations similar to those in section 4, but with equation (19) used for the response time, and the remaining equations modified suitably for multiple classes (see [7] [8] for details). We use three different networks for the validation experiments. For each network, we compute the throughput for different service times, and for each service time setting we evaluate various customer populations. The networks are: 2-station 2-class, 3-station 4-class and 4-station 6-class. We assume probabilistic routing for the customers. For the 2-station network, we set $C_1 = 1$, $C_2 = 2$ for one configuration and $C_1 = 1$, $C_2 = 4$ for another. For the 3-station network, we set $C_1 = C_2 = 1$, $C_3 = 2$ for one configuration and $C_1 = 1$, $C_2 = 2$, $C_3 = 3$ for another. For the 4-station network, we set $C_1 = C_2 = C_3 = 1$, $C_4 = 2$ for one configuration, $C_1 = C_3 = 1$, $C_2 = 3$, $C_4 = 2$ for another, and $C_1 = 1$, $C_2 = 2$, $C_3 = 3$, $C_4 = 4$ for the third. We obtain throughput by solving MS-AMVA non-linear equations using the GAMS equations solver. We also model the networks using the Arena simulation software. The 95% confidence intervals in the simulations were found to be within 2% of the observed mean for all cases. For the 2-station 2-class networks, the MS-AMVA throughput prediction errors were less than 6.5%. For the other two networks the errors were within 10%. Table 3 shows results for a particular 3-station 4-class network configuration.

Table 3: Comparison results for a 3-station 4-class network with $C_1 = 1$, $C_2 = 2$ and $C_3 = 3$

$\vec{N}$	% Utilization			Throughput - Class 1		Throughput - Class 2		Throughput - Class 3		Throughput - Class 4	
	S_1	S_2	S_3	Sim.	MS-AMVA %Error	Sim.	MS-AMVA %Error	Sim.	MS-AMVA %Error	Sim.	MS-AMVA %Error
2,2,2,2	64.2	82.9	81.5	0.2681	-3.8	0.5468	-8.0	0.2748	-5.0	0.3909	-6.6
3,1,4,2	64.8	58.1	91.9	0.2901	-4.2	0.357	-6.4	0.3639	-6.8	0.2469	-8.1
1,1,1,1	45.4	67.4	56.5	0.2018	-1.9	0.429	-7.9	0.1579	0.7	0.3261	-6.8
1,2,3,1	72.1	69.3	81.4	0.1415	-3.2	0.6336	-8.6	0.3886	-6.1	0.2245	-7.3
2,2,2,2	94.1	59.1	77.4	0.1788	-1.0	0.4043	-4.8	0.2906	-5.0	0.4231	0.2
3,1,4,2	90.8	52.9	88.8	0.2431	-0.9	0.2377	-8.3	0.4872	-5.6	0.3333	-6.4
1,1,1,1	84.1	39.8	53.0	0.1237	-2.2	0.3915	-5.7	0.2424	-5.5	0.2467	0.1
1,2,3,1	96.2	35.7	58.8	0.0884	-1.6	0.3667	-5.9	0.4113	-2.5	0.2440	0.8

7. Conclusion

It is evident from the results that the MS-AMVA throughput predictions are reasonably accurate. We conclude that the MS-AMVA method developed using single class networks also provides reasonable throughput estimates for multi-class networks.

References

1. 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), 248-260.
2. Buzen, J.P., 1973, “Computational Algorithms for Closed Queuing Networks with Exponential Servers”, Commun. ACM, 16(9), 527-531.
3. Reiser, M., and Lavenberg, S. S., 1980, “Mean Value Analysis of Closed Multi-chain Queuing Networks”, J.ACM, 27(2) 313-322.
4. Sevcik, K.C., and Mitrani, I., 1979, “The Distribution of Queuing Network States at Input and Output Instants”, in: Performance of Computer Systems, North-Holland, New York, 319-335.
5. Neuse, D., and Chandy K., 1981, “SCAT: A Heuristic Algorithm for Queuing Network Models of Computing Systems”, ACM SIGMETRICS Conf. Proc., 10(3), 59-79.
6. Akyildiz, I.F., and Bolch, G., 1988, “Mean Value Analysis Approximation for Multiple Server Queuing Networks”, Performance Evaluation, 8, 77-91.
7. Schweitzer, P., 1979, “Approximate Analysis of Multi-class Closed Networks of Queues”, Presented at the International Conference, Stochastic Control and Optimization, Amsterdam, The Netherlands.
8. Bard, Y., 1979, “Some Extensions to Multi-class Queuing Network Analysis”, Performance of Computer systems, (M. Arato, ed.), North Holland, Amsterdam.
9. Sakasegawa, H., 1977, “An Approximation Formula Lq ~ $\alpha.\rho^{\beta}$/ (1-ρ)”, Ann. Inst. Statist. Math., 29 A, 67-75.