The inconsistency index method for estimating the accuracy of Schweitzer's approximation

by Dinkar Sitaram

Product-form queueing networks have proved to be useful for predicting the performance of computer systems. In practice, these networks are analyzed using approximate methods because exact methods are computationally too expensive. Schweitzer's approximation is one of the most commonly used. However, there is no method for estimating the error in the solution. This paper proposes a new approach for estimating the error in Schweitzer's approximation for fixed-rate product-form networks. It is based on detecting the extent to which the approximation assumptions used hold. Empirical evidence is presented to show that this approach can be used to accurately predict the error in the approximation.

Introduction

Closed queueing networks have proved to be very useful in analyzing the performance of computer systems. Their use grew after the discovery of fast analysis methods [1, 2] for product-form networks. Hence product-form networks have been used in modeling tools such as VMPPF [3] and RESQ [4].

However, the analysis of large product-form networks or networks with a large number of customer classes by exact methods is still too slow to be usable in practice. Such networks frequently arise in models of computer systems. Hence, a number of approximate methods have been developed. The most widely used methods for closed queueing networks are Schweitzer's approximation [5] and its refinements and extensions [6–8]. These approximation methods are used, for example, in [3, 4].

In spite of the popularity of Schweitzer's approximation, few analytical results about its behavior are available. It has been shown in [9] that there is always a physically meaningful solution which could be returned by the approximation. The convergence of the approximation to this solution and its uniqueness have been shown only for single-class networks in [10]. The approximation is also known to be asymptotically accurate as the population increases to infinity. From experience with the method, it is known that it is generally accurate, and normally has errors of less than 20%. However, there is no systematic method for estimating the error in the solution.

This paper proposes a method for estimating the errors of Schweitzer's algorithm. The proposed method differs from the conventional method of comparing the approximate solution to the exact solution. Instead it relies on the fact that Schweitzer's approximation, like many other approximations, uses "approximation assumptions" to reduce the cost of analysis. Since the approximation assumptions do not hold in general, they are inconsistent with the set of equations defining the exact solution. The proposed method tries to detect the extent of the inconsistency and to use this to predict the error in the solution.

To this end, this paper defines an index for each class at each center, called the inconsistency index. The inconsistency indexes will all be zero if the approximation assumptions are exactly true and the error in the approximate solution is zero. The value of the inconsistency indexes can be computed from the solution returned by the approximation.

The error in Schweitzer's approximation can be predicted from the inconsistency indexes by the use of equations called error equations. It is shown that there are different error equations for different classes of networks, and that the error equation for any class can be found empirically from a sample of networks belonging to that class.

The proposed method also does not require a great deal of extra computation. The computational effort required is $O(SK)$, where S is the number of service centers and K is the number of classes. The inconsistency indexes can be computed without any iterations in one loop over all the centers and classes.

The rest of the paper contains an outline of the proposed method and empirical results to verify its accuracy. First, the notion of an inconsistency index is discussed in general. Then the error measures used in this paper are defined. The proposed method is then described. First, inconsistency indexes for Schweitzer's approximation are derived. Then error equations for various classes of networks are derived and empirical results are presented to show that these equations accurately predict the error in Schweitzer's approximation.

In the sections on Schweitzer's approximation, closed product-form networks are considered since these are widely used in performance modeling. Furthermore, only fixed-rate product-form centers (infinite-server, processor-sharing, or first-come-first-served) are considered. Work is required to see if the method can be generalized to variable-rate centers and other approximate methods.

Inconsistency indexes

Before a formal definition of inconsistency indexes is given, the notion of an inconsistency index is first discussed informally. The discussion does not refer in particular to Schweitzer's approximation. This is because it is applicable to all approximations; i.e., it should be possible to define inconsistency indexes for any approximation. However, further work is necessary to determine if these indexes are useful in predicting the errors of other approximations.

Consider any queueing network where the exact solution is described by a set of "exact equations" Q_e. In order to reduce the complexity of the analysis, a set of "approximation assumptions" Q_a may be used to derive an approximate solution. Typically, in these approximation assumptions the existence of certain relationships among certain quantities in the exact equations is assumed. In general, the approximation assumptions will not hold and will also be inconsistent with the exact equations. On intuitive grounds, it seems that the error in the approximate solution should be related to the extent to which the approximation assumptions do not hold. It also seems reasonable that it should be possible to estimate the error from the lack of fit of the approximation assumptions.

A measure of the lack of fit of the approximation assumptions can be derived in the following manner. Let Y be some quantity in the queueing network, such as the mean queue length. Using some subset of the exact equations and the approximation assumptions, it may be possible to show that $Y = f_1$, where f_1 is some expression involving quantities in the queueing network. Since the exact equations and the approximation assumptions are inconsistent, it may be possible to find a different subset of equations that implies that $Y = f_2$, where in general $f_1 \neq f_2$. Therefore, in general $f = f_1 - f_2$ will not be zero. However, f will be zero if the network parameters are such that the approximation assumptions hold exactly. Furthermore, in many cases, it may be reasonable to assume that if f is close to zero, the approximation assumptions almost hold.

Thus, if an expression f as above can be found, in a sense it measures the extent to which the approximation assumptions hold. It may also be useful in estimating the error in the approximate solution.

In estimating the error, it may be appropriate to normalize f by dividing it by some factor. Such a normalized index will be referred to as an inconsistency index.

The following is a formal definition of inconsistency indexes motivated by the above discussions.

Inconsistent expression Consider any queueing network with an exact solution defined by a set of equations Q_e and analyzed using approximation assumptions Q_a. Let Y be some quantity in the network or some expression involving quantities in the network, and let f, f_1, f_2 be expressions such that

1. Some subset of the equations Q_e and Q_a implies that $Y = f_1$.

2. Some subset of the equations Q_e and Q_a also implies that $Y = f_2$.
3. The equations Q_e by themselves imply that in general $f = f_1 - f_2 \neq 0$.

f will be called an inconsistent expression (*asangata sūtra*)* for the equations Q_e and Q_a associated with the quantity or expression Y.

Inconsistency index Let f be any inconsistent expression for the equations Q_e and Q_a associated with Y, and $M(f)$ be any normalizing factor for f. Then $f/M(f)$ will be called an inconsistency index (*asangati sanketaka*) for the equations Q_e and Q_a associated with Y.

It is to be noted that the above definitions do not furnish any method of finding an inconsistency index. Also, it can be seen that there may be many inconsistency indexes for a given set of equations. In general, experimentation and empirical studies will be required to find useful inconsistency indexes.

Error measures

The error measures studied are the "tolerance" for each class and the maximum tolerance as defined in [11]. This is because the tolerance is a useful measure of the accuracy of an approximation that scales each error in proportion to its importance and does not exaggerate the importance of errors in insignificant quantities. It is found by first normalizing the errors in important performance measures in the network. Queue-length errors are normalized by dividing the queue-length error by the network population. Wait-time errors are normalized by dividing the wait-time error by the total delay for one trip around the network. Thus each error is scaled in proportion to its importance, and large errors in insignificant quantities are not magnified. The tolerance is defined as the largest normalized error.

Mathematically, the queue-length tolerance error for any class c at any center s is

$$E_{L,sc} = \left| \frac{L_{sc}^{(ex)} - L_{sc}^{(ap)}}{N_c} \right|,$$

where $L_{sc}^{(ex)}$ and $L_{sc}^{(ap)}$ are the exact and approximate values for the queue length. The wait-time tolerance error is

$$E_{W,sc} = \left| \frac{W_{sc}^{(ex)} - W_{sc}^{(ap)}}{\sum_{s=1}^{S} W_{sc}^{(ex)}} \right|,$$

where $W_{sc}^{(ex)}$ and $W_{sc}^{(ap)}$ are the exact and approximate values for the mean wait time per trip around the network. The tolerance error for class c, E_c, is given by

$$E_c = \max_s (E_{L,sc}, E_{W,sc}).$$

The maximum tolerance error is

$$E = \max E_c.$$

*Here and in a few other instances, we introduce some technical terms into Sanskrit.

Indexes for Schweitzer's approximation

• *Approximation assumptions*

For clarity, the approximation assumptions used in Schweitzer's approximation are reviewed before the inconsistency indexes are derived. Consider a closed product-form queueing network containing S fixed-rate centers and K classes. Let the population vector in the network be N, with the population in class k being denoted by N_k. Let D_{sk} be the loading of class k at center s (the product of the visit ratio and mean service time for class k at center s). Let the mean queue length, wait time, and utilization when the population vector is n be denoted by $L_{sk}(n)$, $W_{sk}(n)$, $\rho_{sk}(n)$, respectively. Let e_k be a K-component vector whose kth component is 1 and whose other components are 0. Schweitzer's approximation sets

$$L_{sc}(N - e_k) = \begin{cases} L_{sc}(N) & \text{if } c \neq k, \\ \dfrac{N_k - 1}{N_k} L_{sc}(N) & \text{if } c = k. \end{cases} \tag{1}$$

The traditional motivation for this assumption is that it is equivalent to assuming that the change in queue lengths caused by removing one customer from the network is distributed over the centers in proportion to their queue lengths. However, as pointed out in [10], there is an alternative motivation for this assumption. For networks with infinite-server centers, relation (1) is equivalent to assuming that the mean wait time at each center at population level $N - 1$ is the same as the mean wait time at population level N, i.e., that

$$W_{sc}(N - e_k) = W_{sc}(N). \tag{2}$$

For networks with no infinite-server centers, relation (2) is a sufficient but not necessary condition for relation (1). In the following, the assumptions behind Schweitzer's approximation will be strengthened somewhat by assuming that the approximation uses relation (2) even when there are no infinite-server centers. As shown later, relation (2) is necessary to compute the inconsistency indexes. Relations (1) and (2) then form the approximation assumptions used in Schweitzer's approximation and will be referred to in what follows as "the approximation assumptions."

• *Derivation of inconsistency indexes*

The inconsistency indexes for Schweitzer's approximation are defined by noting that the approximation assumptions together with the mean value analysis equations imply that there are two distinct ways of computing $L_s(N - e_k)$ at the noninfinite-server centers. There is thus an inconsistency index associated with the queue length for Schweitzer's approximation. First, from approximation assumption (1),

$$L_s^{(1)}(N - e_k) = L_s(N) - \frac{1}{N_k} L_{sk}(N). \tag{3}$$

Second,

$$L_s^{(2)}(N - e_k) = \frac{u_s(N - e_k)}{1 - u_s(N - e_k)}, \tag{4}$$

where

$$u_s(N - e_k) = \sum_{c=1}^{K} \frac{(N - e_k)_c \rho_{sc}(N - e_k)}{(N - e_k)_c + \rho_{sc}(N - e_k)}. \tag{5}$$

The detailed derivation of these equations is given in the appendix. For large N, it can be seen that u_s tends to ρ_s, the utilization of server s. It can be seen that in general $L_s^{(1)}(N - e_k) \neq L_s^{(2)}(N - e_k)$. $L_s^{(1)}(N - e_k) - L_s^{(2)}(N - e_k)$ is thus an inconsistent expression. The queue-length inconsistency index is defined as

$$I_{skL}(N) = \frac{L_s^{(1)}(N - e_k) - L_s^{(2)}(N - e_k)}{\sum_{c=1}^{K} N_c}. \tag{6}$$

Because the average wait time is given by $W_{sk}(N) = D_{sk}(1 + L_s(N - e_k))$ and there are two ways of computing $L_s(N - e_k)$, it can be seen that there are two distinct ways of computing $W_{sk}(N)$. These are

$$W_{sk}^{(1)}(N) = D_{sk}(1 + L_s^{(1)}(N - e_k)) \tag{7}$$

and

$$W_{sk}^{(2)}(N) = D_{sk}(1 + L_s^{(2)}(N - e_k)). \tag{8}$$

The wait-time inconsistency index is defined as

$$I_{skW}(N) = \frac{W_{sk}^{(1)}(N) - W_{sk}^{(2)}(N)}{W_k(N)}, \tag{9}$$

where

$$W_k(N) = \sum_{s=1}^{S} W_{sk}(N) \tag{10}$$

is the total delay for class k for one trip around the network.

In order to compute $L_s^{(2)}(N - e_k)$ from Equation (4), it is necessary to compute $\rho_{sc}(N - e_k)$. Approximation assumption (2) implies that

$$\rho_{sc}(N - e_k) = \begin{cases} \rho_{sc}(N) & \text{if } c \neq k, \\ \dfrac{N_k - 1}{N_k} \rho_{sc}(N) & \text{if } c = k. \end{cases} \tag{11}$$

The following equations can be used to more efficiently compute the inconsistency indexes. Let

$$u_s(N) = \sum_{c=1}^{K} \frac{N_c \rho_{sc}(N)}{N_c + \rho_{sc}(N)}. \tag{12}$$

Then

$$u_s(N - e_k) = u_s(N) - \frac{N_k \rho_{sk}(N)}{N_k + \rho_{sk}(N)} + \frac{(N_k - 1)\rho_{sk}(N - e_k)}{N_k - 1 + \rho_{sk}(N - e_k)}. \tag{13}$$

The algorithm for computing the inconsistency indexes thus is

1. Run Schweitzer's approximation to get estimates of the performance measures.
2. Repeat steps 3 and 4 for all processor-sharing or FCFS centers s.
3. Compute $u_s(N)$ from Equation (12).
4. For all classes k compute $u_s(N - e_k)$ from Equations (13) and (11); compute $L_s^{(1)}(N - e_k)$ from Equation (3) and $L_s^{(2)}(N - e_k)$ from Equation (4); compute $W_s^{(1)}(N)$ from Equation (7) and $W_s^{(2)}(N)$ from Equation (8); compute the inconsistency indexes $I_{skL}(N)$ and $I_{skW}(N)$ from Equations (5) and (9).

Error equations

The tolerance error in Schweitzer's approximation can be predicted from the inconsistency indexes by the use of error equations (*bheda sūtra*). Error equations are developed below for two classes of networks. The characteristics of these networks are described in greater detail later. The first class is a class of unsaturated networks. The second class is a class of saturated networks with high errors found in [12]. Similar error equations can be found for other classes of networks.

The following quantities have been found to be useful for developing error equations. They are

$$\hat{I}_{kL} = \max_s |I_{skL}|,$$

$$\hat{I}_{kW} = \max_s |I_{skW}|,$$

the maximum value of the queue-length and wait-time inconsistency indexes,

$$\bar{I}_{kL} = \sum_{s=1}^{S} I_{skL},$$

$$\bar{I}_{kW} = \sum_{s=1}^{S} I_{skW},$$

the sum of the inconsistency indexes over all the servers, and

$$\bar{\bar{I}}_{kL} = \sqrt{\sum_{s=1}^{S} I_{skL}^2},$$

$$\bar{\bar{I}}_{kW} = \sqrt{\sum_{s=1}^{S} I_{skW}^2},$$

the root sum of the squares of the inconsistency indexes.

- *Unsaturated networks*

The first class of networks for which an error equation is developed is a class of unsaturated networks. The number of centers in the networks was uniformly distributed between 2 and 30. The total population in the network was allowed to vary uniformly from 2 to twice the number of centers. The restriction on the population was imposed as otherwise the

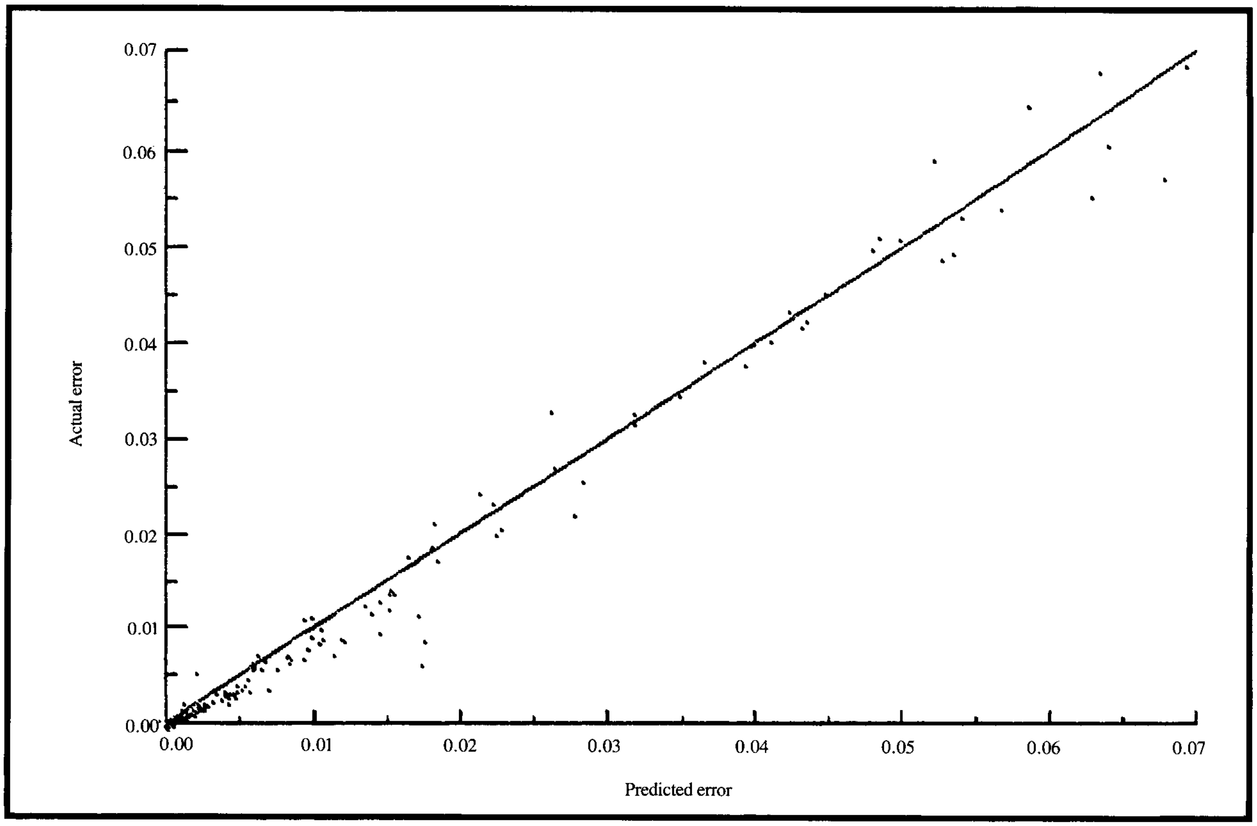

Figure 1

Predicted error vs actual error for single-class networks.

resulting networks frequently contained at least one center which was heavily saturated. For multiple-class networks, the total population was divided up into a number of customer classes. This was done by repeatedly allocating a random number of customers between 3 and the total number of centers to classes until the total population had been allocated.

The loadings at the centers were allowed to vary between 1 and 10. Since all the infinite-server centers in any product-form network can be aggregated into a single center, the first center was always an infinite-server center. The other centers were processor-sharing centers, since in a product-form network the FCFS centers can be replaced by processor-sharing centers without changing the solution.

An error equation for these networks was found by experimenting with a random set of 150 such single-class networks. This error equation was then verified by using it to predict the errors in a set of randomly generated two-class, three-class, and four-class networks. The error equation was found to predict fairly well both the per-class tolerance error and the maximum tolerance error of the multiple-class networks.

The error equation was found by regressing the error for single-class networks against the quantities $\hat{I}_{kL}$, $\hat{I}_{kW}$, $\overline{T}_{kL}$, $\overline{T}_{kW}$, $\overline{\overline{T}}_{kL}$, and $\overline{\overline{T}}_{kW}$. It was found that the equation

$$E_k = 0.94\hat{I}_{kL} - 0.074\overline{T}_{kL} - 0.72\overline{T}^2_{kL} - 1.55\hat{I}_{kW}\overline{T}_{kW} \qquad (14)$$

provides a good fit to the tolerance errors of the networks.

Figure 1 illustrates the fit of the predicted error P to the actual error E for single-class networks. The straight line is the line $P = E$. The actual error for each network was plotted against the predicted error. If all the predicted errors were equal to the actual errors, all the points would lie on the line $P = E$. At the points lying above the line $P = E$, the predicted error was less than the actual error. At the points under the line, the predicted error was greater than the actual error. It can be seen that at most of the points, the predicted error is very close to the actual error. The coefficient of correlation between the predicted and actual errors was found to be 0.992.

Figures 2 and **3** are similar plots that illustrate the extent of the fit between the predicted and actual errors for multiple-class networks. The first plot illustrates the fit between the predicted error for each class and the actual

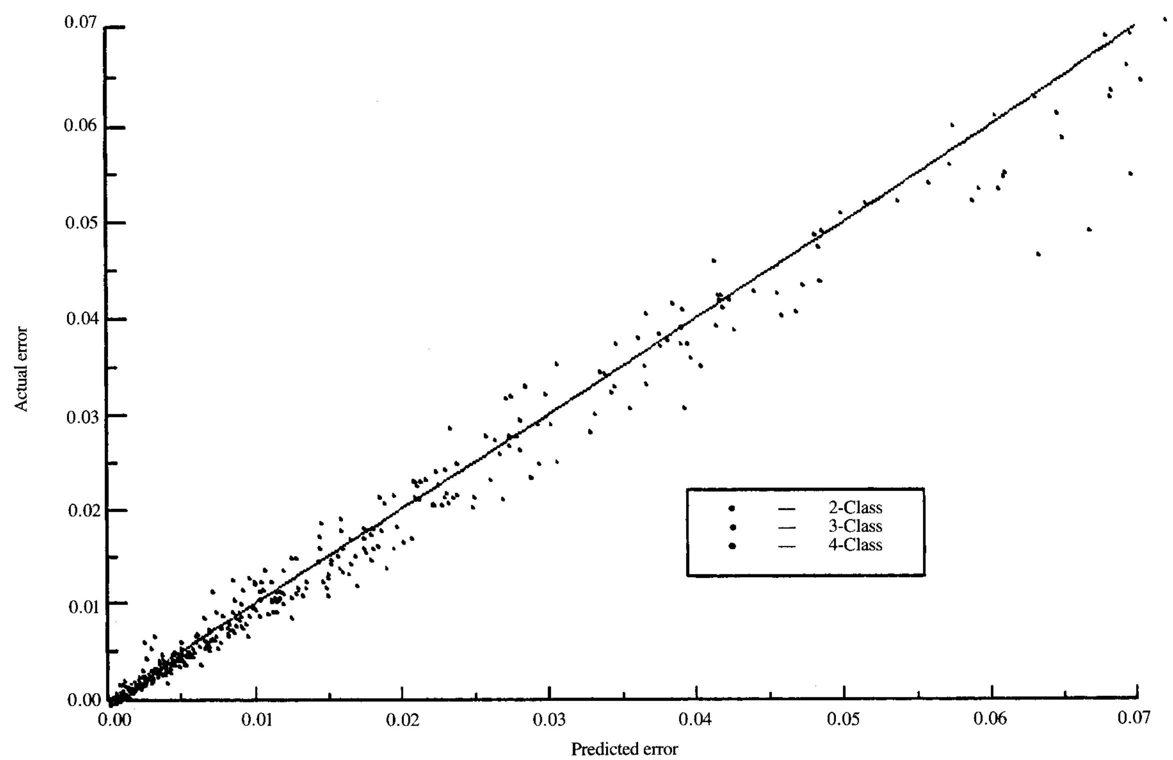

Figure 2

Per-class predicted error vs per-class actual error for multiple-class networks.

error for each class in the networks. The second plot illustrates the fit between the maximum predicted error over all classes and the maximum actual error over all classes. It is apparent that the fits for the two-, three-, and four-class networks are indistinguishable. This is borne out by the data in **Table 1**.

Table 1 shows the correlation between the predicted and actual errors for the multiple-class networks. The table shows, for example, that the best (least-squares) estimate of the error for two-class networks is given by $E = 1.005P$ and that the coefficient of correlation between E and P for two-class networks is 0.997. The error equation predicts both the error for each class and the maximum error fairly well.

Table 1 Predicted vs actual errors for multiple-class networks.

Number of classes	*Per-class*		*Maximum*	
	Coeff.	*Corr.*	*Coeff.*	*Corr.*
2	1.005	0.997	1.019	0.998
3	0.960	0.993	0.968	0.996
4	1.000	0.994	1.000	0.997

• *Saturated high-error networks*

The second class of networks for which an error equation is developed is a class of saturated high-error networks from [12]. Each network has two processor-sharing centers and two customer classes. Both classes visit the first center; the second center is visited only by class 2. The class loadings at the first center are 1 and $1/(2 + \varepsilon)$; the loading for class 2 at the second center is $(1 + \varepsilon)/(2 + \varepsilon)$, where ε is some constant varying between 0 and 1. There is one customer of class 1; the number of customers in class 2 is variable.

It is shown in [12] that as the number of class 2 customers approaches infinity, the wait-time tolerance error for class 1 approaches $1/(1 + \varepsilon)$. In particular, as $\varepsilon \rightarrow 0$, the tolerance error tends to 100%. The tolerance errors for class 1 are small. An error equation for this class of networks was found by generating a set of 81 networks. The population in class 2 was varied from 10 to 90 in steps of 10; ε was varied from 0.1 to 0.9 in steps of 0.1. Using regression analysis, it was found that the error for class 1 was predicted by the equation

$$E_1 = 1.16\overline{I}_{1L} + 1.07\overline{I}^2_{1L} - 1.4\overline{I}_{1W},$$

and that the error for class 2 was predicted by the equation

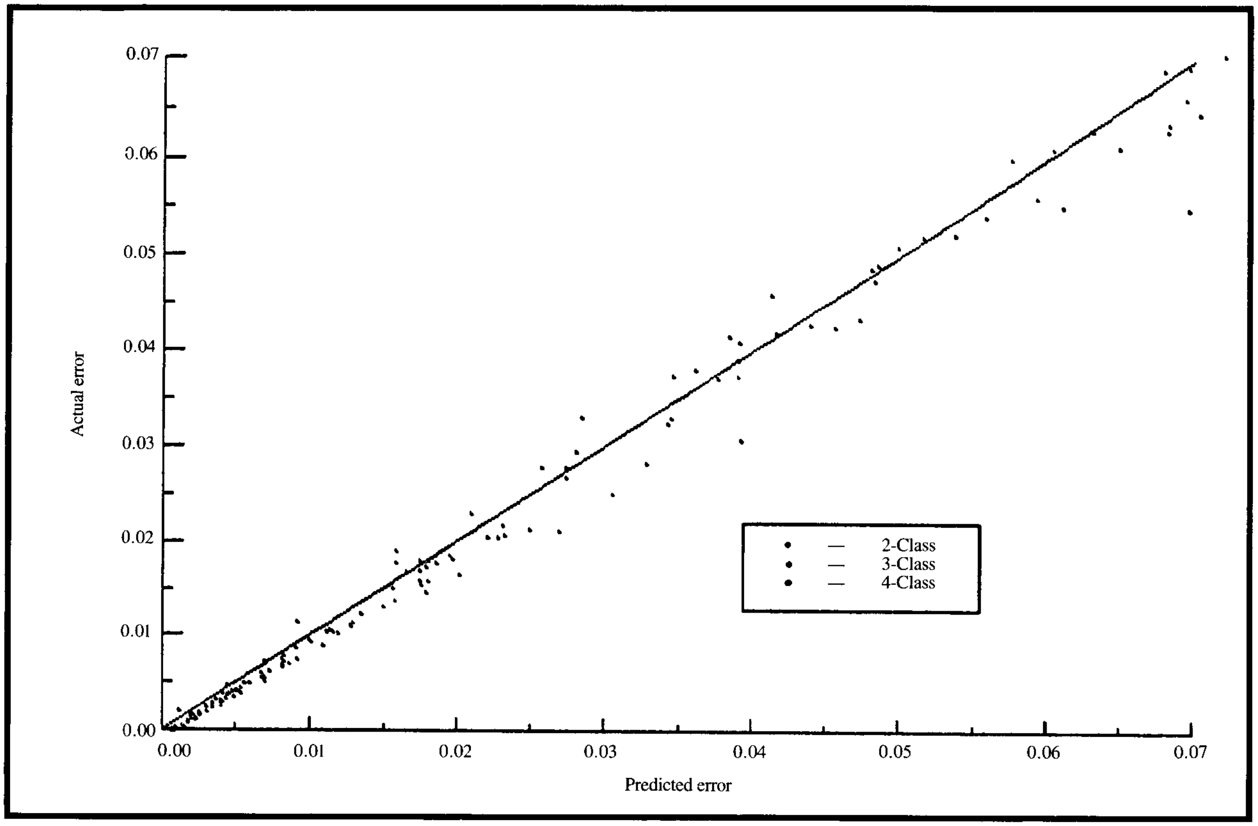

Figure 3

Maximum predicted error vs maximum actual error for multiple-class networks.

$$E_2 = 2.68\bar{\bar{I}}_{2L} - 3.13\bar{\bar{I}}_{2W} + \bar{\bar{I}}^2_{2W} - 3.23\bar{I}^2_{2L} + 5.191\bar{I}^2_{2W}.$$

Figure 4 illustrates the fit of this equation to the actual errors. It can be seen that there is a good fit between the predicted and actual errors.

Computational complexity

From the section describing the computation of the inconsistency indexes, it can be seen that the inconsistency indexes can be computed in one pass over all the centers. At each pass, two loops have to be made over all the classes. Thus it requires $O(SK)$ computation time to compute the inconsistency indexes. Clearly the computation of the quantities $\hat{I}$, $\bar{I}$, and $\bar{\bar{I}}$ that are used to estimate the error also requires $O(SK)$ time. In fact, the computation can be integrated into step 4 of the computation of the inconsistency indexes. Thus the overall complexity of the method is $O(SK)$. Since no iterations are necessary, the computation required by the method is small compared to the computation required by Schweitzer's approximation.

Conclusions

This paper has proposed a new approach to estimating the errors of Schweitzer's approximation. The approach is based on detecting inconsistencies in the approximation assumptions made in deriving the approximation. Error equations are then empirically found that relate the error to the extents of the inconsistencies. Empirical evidence has been presented to show that this approach works satisfactorily for Schweitzer's approximation with fixed-rate servers. Additional work is necessary to see if this approach can be extended to variable-rate servers and other approximations.

Acknowledgments

I would like to thank R. M. Bryant for his perceptive comments about the organization of the paper, and Y. Bard and the anonymous referees for their helpful comments. I would particularly like to thank D. L. Eager for pointing out that my original definition of the wait-time inconsistency index was wrong and for his many encouraging comments and suggestions. I would also like to thank J. Weglarz,

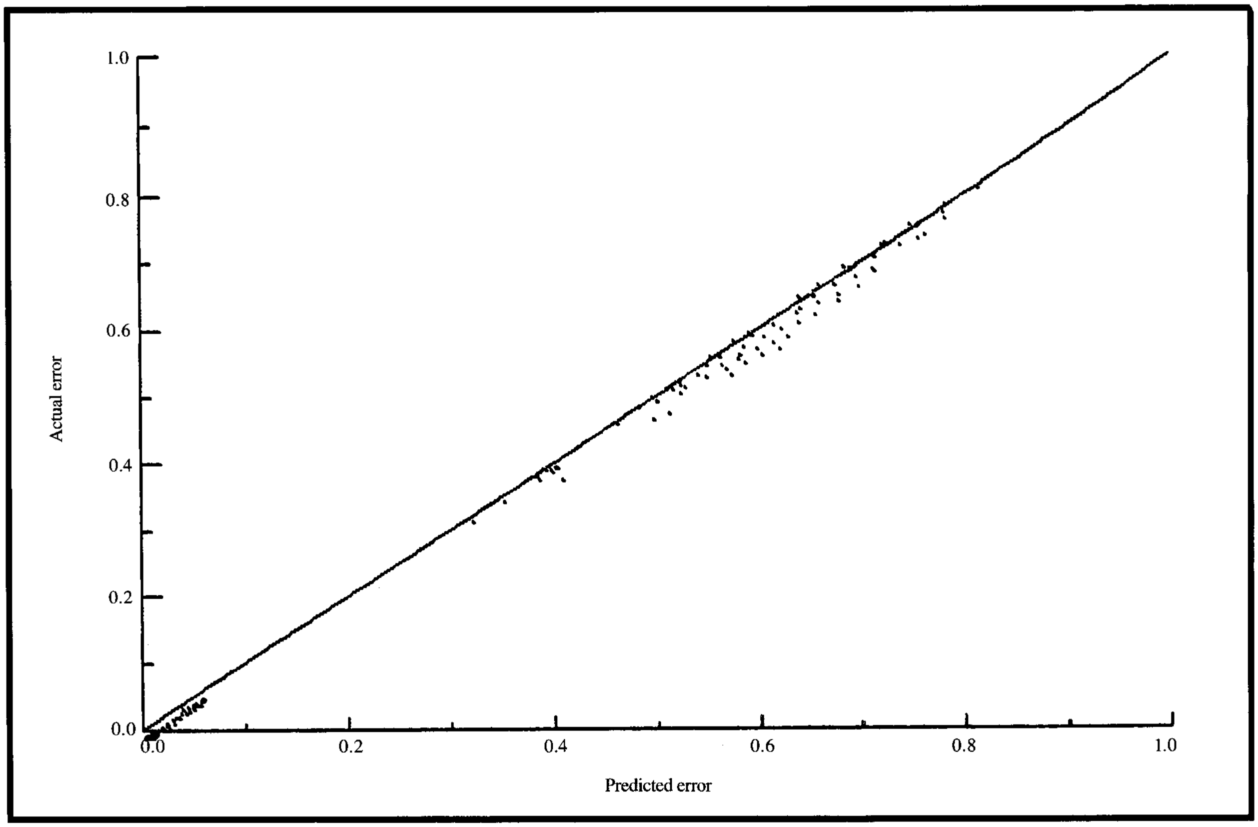

Figure 4

Predicted error vs actual error for saturated high-error networks.

T. Deuser, A. Lipton, and C. Lipton for their help in producing this paper.

Appendix

This appendix derives some of the formulae assumed in the paper. Relation (3),

$$L_s(N - e_k) = L_s(N) - \frac{1}{N_k} L_{sk}(N),$$

clearly follows directly from approximation assumption (1):

$$L_{sc}(N - e_k) = \begin{cases} L_{sc}(N) & \text{if } c \neq k, \\ \dfrac{N_k - 1}{N_k} L_{sc}(N) & \text{if } c = k. \end{cases}$$

It is now shown that

$$L_s(n) = \frac{u_s(n)}{1 - u_s(n)}, \tag{A1}$$

where

$$u_s(n) = \sum_{c=1}^{K} \frac{n_c \rho_{sc}(n)}{n_c + \rho_{sc}(n)}, \tag{A2}$$

which implies relation (4). First, from relation (1),

$$L_s(n - e_k) = L_s(n) - \frac{1}{n_c} L_{sc}(n).$$

Substituting this into the mva equation for wait time gives

$$W_{sc}(n) = D_{sc}\left(1 + L_s(n) - \frac{1}{n_c} L_{sc}(n)\right).$$

Multiplying both sides by the throughput $\lambda_{sk}(n)$ and using Little's rule gives

$$L_{sc}(n) = \rho_{sc}(n)\left(1 + L_s(n) - \frac{1}{n_c} L_{sc}(n)\right).$$

Rearranging the above equation gives

$$L_{sc}(n) = \frac{n_c \rho_{sc}(n)}{n_c + \rho_{sc}(n)} (1 + L_s(n)).$$

Summing the above equation over all classes and noting that the left-hand-side sum $\sum_{c=1}^{K} L_{sc}(n)$ reduces to $L_s(n)$ gives

$$L_s(n) = \sum_{c=1}^{K} \frac{n_c \rho_{sc}(n)}{n_c + \rho_{sc}(n)} (1 + L_s(n)).$$

Rearranging the above equation gives relations (A1) and (A2). For the case where there is only one class, this reduces to the equation

$$L_s(n) = \frac{\rho_s(n)}{1 - (n - 1)/n\rho_s(n)}.$$

References

1. J. P. Buzen, "Queueing Network Models of Multiprogramming," Ph.D. thesis, Harvard University, Cambridge, MA, 1971.
2. M. Reiser and S. S. Lavenberg, "Mean Value Analysis of Closed Multichain Queueing Networks," *J. ACM* **27,** 313–322 (April 1980).
3. Y. Bard, "The VM/370 Performance Predictor," *Comput. Surv.* **10,** No. 3, 333–342 (September 1978).
4. Charles H. Sauer, Edward A. McNair, and Silvio Salza, "A Language for Extended Queuing Network Models," *IBM J. Res. Develop.* **24,** No. 6, 747–755 (November 1980).
5. P. Schweitzer, "Approximate Analysis of Multiclass Closed Networks of Queues," *Proceedings of the International Conference on Stochastic Control and Optimization,* Amsterdam, 1979.
6. K. M. Chandy and D. Neuse, "Linearizer: A Heuristic Algorithm for Queueing Network Models of Computing Systems," *Commun. ACM* **25,** No. 2, 126–134 (1982).
7. A. Krzesinski and J. Greyling, "Improved Lineariser Methods for Queueing Networks with Queue Dependent Centres," *Proceedings of the ACM Sigmetric Conference on Measurement and Modeling of Computer Systems,* 1984, pp. 41–51.
8. J. Zahorjan and E. D. Lazowska, "Incorporating Load Dependent Servers in Approximate Mean Value Analysis," *Proceedings of the ACM Sigmetric Conference on Measurement and Modeling of Computer Systems,* 1984, pp. 52–62.
9. E. deSouza e Silva, S. S. Lavenberg, and R. R. Muntz, "A Perspective on Iterative Methods for the Approximate Analysis of Closed Queueing Networks," *Mathematical Computer Performance and Reliability,* G. Iazeolla, P. J. Courtois, and A. Hordjik, Eds., North-Holland Publishing Co., Amsterdam, 1984, pp. 225–244.
10. D. L. Eager and K. C. Sevcik, "An Analysis of an Approximation Algorithm for Queueing Networks," *Perform. Eval.* **4,** 275–284 (1984).
11. D. Neuse and K. Chandy, "A Heuristic Algorithm for Queueing Network Models of Computer Systems," *Report TR-CHAN-81-02,* University of Texas at Austin, March 1981.
12. D. L. Eager, "Bounding Algorithms for Queueing Network Models of Computer Systems," *Technical Report CSRG-156,* University of Toronto, March 1984.

Received February 14, 1986; accepted for publication December 5, 1986

Dinkar Sitaram *IBM Data Systems Division, P.O. Box 100, Kingston, New York 12401.* Dr. Sitaram received his B.Tech. from the Indian Institute of Technology, Kharagpur, India, and his M.S. and Ph.D. from the University of Wisconsin, Madison, in 1985. He is currently working in the VM Advanced Systems Design Department at Kingston. His interests lie in the field of computer performance evaluation, particularly in the analytic modeling of computer systems.