RESPONSE TIME PRESERVATION:

A General Technique for Developing Approximate Algorithms for Queueing Networks

Subhash C. Agrawal, Jeffrey P. Buzen, Annie W. Shum

BGS Systems, Inc.

One University Office Park
Waltham, MA. 02254

ABSTRACT

Response Time Preservation (RTP) is introduced as a general technique for developing approximate analysis procedures for queueing networks. The underlying idea is to replace a subsystem by an equivalent server whose response time in isolation equals that of the entire subsystem in isolation. The RTP based approximations, which belong to the class of decomposition approximations, can be viewed as a dual of the Norton's Theorem approach for solving queueing networks since it matches response times rather than throughputs. The generality of the RTP technique is illustrated by developing solution procedures for several important queueing systems which violate product form assumptions. Examples include FCFS servers with general service times, FCFS servers with different service times for multiple classes, priority scheduling, and distributed systems.

1. Introduction

Queueing network models have been found to be extremely useful and cost-effective in analyzing the performance of complex computer systems. The wide applicability of these models is due primarily to the discovery of efficient computational algorithms [Buzen 73, Bruell and Balbo 80, Reiser and Lavenberg 80] for product-form queueing networks [Basket et. al. 75]. Many real systems, however, exhibit characteristics that violate the product form assumptions. Typical examples include priority scheduling at a server, queueing for passive resources such as critical sections and memory, I/O path contention, database concurrency algorithms, and blocking.

Various approximations have been developed to handle networks with such properties [Agrawal 83a, Bard 79, Bard 80, Brandwajn 74, Brandwajn 82, Courtois 75, Potier and Leblanc 80, Graham 78]. While each approximation may appear to involve an entirely different technique, Agrawal and Buzen have unified their characterization through a general framework termed metamodeling [Agrawal 83b, Buzen and Agrawal 83]. As explained by them, the principal idea in developing an approximation is to transform the original network into one or more simpler networks, solve these simpler networks, and then integrate their solutions to obtain an approximate solution of the original system. Each one of the approximations mentioned above can be viewed as an application of a transformation or a series of such transformations. Transformations for a number of approximation techniques are discussed in [Agrawal 83b].

In this paper we present a general approximation development technique which entails isolating the subsystem from the original model, analyzing the isolated subsystem under an assumed arrival process and replacing the subsystem by an equivalent server whose response time under the assumed arrival process equals that of the isolated subsystem. Since the underlying transformation preserves the isolated system response time, it is called a Response Time Preservation transformation. The resulting approximation procedure is called a Response Time Preservation (RTP) based approximation.

We first motivate the RTP technique by developing an approximation for modeling FCFS servers with general service times in a queueing network. Then we specify the general technique in Section

3 and present some specific mappings in Section 4. These mappings are followed by additional examples in Section 5. Some considerations in developing an effective RTP approximation are discussed in Section 6. Section 7 compares the RTP approach and the Norton's Theorem approach [Chandy et. al. 75] as two decomposition based solution approximation development procedures.

One of the most important aspects of the RTP approximation is that it is a general technique for developing approximate solution procedures for a wide class of non-product form queueing systems. It is not our intent here to evaluate extensively the accuracy of any individual RTP based approximation. Rather, we wish to emphasize the underlying concept and the essential steps of the RTP procedure so that the reader can develop specific RTP based approximations for individual problems. Detailed analyses of the accuracy of specific RTP based approximations is a subject for future research.

2. Example: FCFS Servers with General Service Times

Consider a single class closed queueing network containing a FCFS server as shown in Figure 1a. If the service time at this server is not exponentially distributed, the queueing network does not possess an efficiently computable product form solution. A number of approximation procedures have been proposed for solving such networks [Shum and Buzen 77, Marie 78, Balbo 79].

As an alternative to these previously published approaches, consider the following intuitive technique for solving such networks: Replace the general FCFS server by an equivalent server with exponentially distributed service times. The resulting network, as depicted in Figure 1b, possesses a product form solution and thus is easily solvable.

The primary issue now is to obtain the service times at the equivalent server. For a good approximation, one important condition is that the response time of a job at the general FCFS server should be the same as its response time at the equivalent server. As an approximation to this condition, we equate the response time of a job at the two servers when they are taken out of the system and analyzed by assuming that the arrivals are generated by a Poisson or homogeneous arrival process. See Figure 1c.

The next step is to compute the two response times. If the general FCFS server has throughput equal to X and service time equal to S with a coefficient of variation CV, then by the Pollaczek-Khinchin formula for an M/G/1 queue with FCFS scheduling [Kleinrock 75], the open system response time is given by:

$$R = S + \frac{(1 + CV^2)\, S\, U}{2(1 - U)}, \qquad (1)$$

where $U = X S$ is the server utilization.

Let the service time at the exponential equivalent server be S'. Then, the response time at this server is given by the standard M/M/1 formula:

$$R' = \frac{S'}{1 - X S'}.$$

Our response time preservation technique requires that

$$R' = R.$$

Solving for S1 and substituting R' for R, the effective service time at the equivalent server is:

$$S' = \frac{R}{1 + X R}. \qquad (2)$$

The complete solution procedure for the closed queueing network is iterative and outlined below:

1. Assume initial throughput X'.
2. Repeat
   X = X'
   For general server i
   Compute Ri using eq. (1).
   Compute Si' using eq. (2).
   Obtain new throughput X' of the product form network containing the equivalent server by using MVA or Convolution.
   Until $|X' - X| < \epsilon$.

Note that, in this procedure, we create a product form network of servers whose response times and throughput rates approximate those of the original non-product form network. To calculate server utilizations in the original network, we simply multiply the approximate server throughputs by the original service times.

The data presented in Table 1 shows that the method yields reasonably accurate throughputs for moderate CVs (up to 5). Device utilizations and system response time have comparable accuracy. Errors are larger for higher CVs. When service times are exponential, the method is exact. See Appendix A for a more systematic examination of this particular approximation.

3. Response Time Preservation Approximation

The Response Time Preservation approximation used implicitly in the previous section can be generalized to provide a powerful approximation development technique. To this end, let us abstract the technique from the preceding example.

Start with an original model (Mo) whose parameters (Po) include device service times and whose performance metrics (Qo) include system throughput. Mo is not easily solvable because it contains a non-product form subsystem, namely the FCFS server with general service times. Construct a product-form queueing network (M) by replacing the non-product form subsystem (general server) with an equivalent product form subsystem (an "equivalent" server). The forward mapping F from Mo to M determines the parameters of the equivalent subsystem, e.g., the service time of the equivalent server. F is such that the response time of the non-product form subsystem under an assumed arrival process (in this case Markovian) is equal to the response time of the equivalent product form subsystem under the same arrival process. The reverse mapping R from M to Mo equates Xo, the throughput of Mo, to X which is the throughput of M.

To be able to parameterize the equivalent subsystem, we need to compute the response time of the non-product form subsystem under the assumed arrival process. To accomplish this, construct an auxiliary model M1 representing only the subsystem with the assumed arrival process. The forward mapping F1, from Mo to M1, computes the parameters of the subsystem including the parameters of the arrival process. In the special case of Markovian arrivals, only the mean arrival rate is required. In general, the arrival rate equals the network throughput times the appropriate visit ratio. However, the throughput is not known a priori and is, therefore, iteratively computed.

The structure of an RTP based approximation procedure immediately follows: Let Mo be a model of a system with N subsystems each of which can be analyzed in isolation (say, as an open system). Then the RTP approximation procedure is:

0. Assume initial system throughput Xo.

1. Isolate and solve each subsystem Mi, i=1,...,N

   a. Using forward mapping Fi, compute arrival process parameters,
      e.g. arrival rate at Mi.
   b. Solve Mi in isolation and calculate the
      subsystem's response time Ri.

2. Construct and solve transformed system model M.

   a. F: Using forward mapping F, compute effective service time at the
      equivalent server representing subsystem i.
   b. Solve M (with product form algorithms)
      and compute new throughput Xn.

3. If $|Xn - Xo| < \epsilon$ STOP
   else Xo = Xn
        go to 1

An RTP based approximation can be developed whenever solutions can be obtained for the subsystems in isolation. It is based on the assumption that if the response time of a subsystem in isolation equals the response time of the equivalent server in isolation, then it is likely that the response time at the equivalent server in the transformed model M will equal the response time at the subsystem in the original model Mo. This assumption is reminiscent of the On-line = Off-line behavior implicit in traditional Norton's Theorem based decomposition approximations [Brandwajn 74, Chandy et.al. 75, Courtois 75, Denning and Buzen 78]. We discuss this analogy and compare the two approaches in Section 7. We now present some forward mappings for computing the effective service times at the equivalent server(s) in specific cases.

4. Forward Mappings for Equivalent Servers

One of the crucial steps in developing an RTP based approximation is to characterize the equivalent server representation for a subsystem and to compute a customer's effective service time. The choice of equivalent server(s) is affected by a number of considerations that are detailed in Section 6. One consideration is the nature of the arrival process used in the analysis of the isolated subsystem. If the arrival process is assumed to be Poisson or homogeneous, some of the forward mappings for computing effective service times at the equivalent server(s) are relatively simple. If the arrival process assumed for the isolated subsystem analysis is similar to the one observed at the subsystem in the network, the forward mapping may be trivial. Some of these mappings are presented in terms of the following theorems. The mappings are differentiated by the number of classes and the number of equivalent servers.

Theorem 4.1 - Single Class Equivalent Server:

Assume that the forward mapping Fi used to isolate a single class subsystem yields an open network having Poisson arrivals with rate X. Let the response time of the isolated subsystem under this mapping be R. Then, the effective service time at the response time preserving equivalent server is:

$$S' = \frac{R}{1 + X R} . \qquad (3)$$

Proof:

From equations 1 and 2 of the general FCFS server example.

Corollary 1:

Assume that the forward mapping Fi used to isolate a multi-class subsystem yields an open network in which customer class r has Poisson arrivals with rate Xr for r = 1, ..., c. Let Rr be the response time of customer class r under this mapping. Assume that in the equivalent product form network each customer class is processed by a dedicated equivalent server. Then the effective service time of class r at its equivalent server is given by

$$Sr' = \frac{Rr}{1 + Xr\ Rr}, \quad r = 1, \ldots, c. \qquad (4)$$

An alternative approach for constructing equivalent servers for a multiclass subystem is given by the following theorem.

Theorem 4.2 - Multiple Class Equivalent Server:

Assume that the forward mapping Fi used to isolate a multi-class subsystem yields an open network in which customer class r has Poisson arrivals with rate Xr for r = 1, ..., c. Let Rr be the response time of customer class r under this mapping. Assume that, in the equivalent product form network, all customer classes are processed by a single equivalent server using a processor sharing discipline. The effective service time for class r, Sr', $1 \leq r \leq c$, is given by:

$$Sr' = \frac{Rr}{1 + \sum_{i=1}^{c} Xi\ Ri} . \qquad (5)$$

Proof: Class r response time at the equivalent server is

$$Rr = \frac{Sr'}{1 - \sum_{k=1}^{c} Xk\ Sk'} . \qquad (6)$$

It is easy to verify that the solution given by equation (5) satisfies the above equation. And since eq. (6) represents a system of c equations in c unknowns, the solution of eq. (5) is unique.

When the interarrival times at subsystem Mi are not exponentially distributed, the calculations of effective service times are not necessarily as simple. In general, for a given arrival process at Mi, if the response time function of the equivalent server is

$$Ri' = fi(Si'), \qquad (7)$$

then by the RTP approximation Ri' = Ri, and therefore the effective service time is given by

$$Si' = fi^{-1}(Ri). \qquad (8)$$

If the equivalent server is just a delay server (i.e., an infinite or no-queueing server) then Si' simply equals the response time Ri. It is sometimes desirable to use delay servers when developing RTP approximations. For example, assume that the isolated subsystem can be analyzed under an arrival process that approximates closely the arrival process observed at the subsystem in the original network. Let the isolated subsystem's response time be R. Then, one appropriate forward mapping F is to replace the subsystem by a delay server with delay R.

5. Additional Applications of the RTP Technique

In this section we show how RTP based approximations can be developed for a number of networks that violate product form conditions. Complete equations are presented for RTP approximations of multi-class FCFS servers and priority servers. This is followed by a discussion of distributed systems and a brief outline of how RTP could be used to integrate isolated models of nodes, communication networks and synchronization delays. These examples provide an indication of the broad applicability of the RTP approach.

5.1 Different Service Times at a FCFS Server:

Consider a c-class closed queueing network Mo with an FCFS server, i. In general, the mean and variance of the service time at server i for each customer class is different. In this case, the network does not have a product form solution. An RTP based approximation that is a multi-class generalization of the case treated in Section 2 can be obtained by applying Theorem 4.2. The approximation entails replacing server by a processor sharing equivalent server i' (Model M). To accomplish this, we first need to find the response time for each class at the isolated server (Model Mi).

Model Mi: This model consists of an isolated FCFS server visited by c classes, each with Poisson arrivals and general service times. The arrival rate of class r is VirXr, where Vir is the number of class r visits to server i in Mo and Xr is class r throughput in Mo. The mean and coefficient of variation of the service time are Sir and CVir, respectively. The response time of class r, Rir, is computed as follows.

Total arrival rate at the isolated server

$$Xi = \sum_{r=1}^{c} Vir\ Xr.$$

Mean effective service time at the server is given by

$$Si = \frac{\sum_{r=1}^{c} Sir\ Vir\ Xr}{Xi}.$$

Coefficient of variation square for the effective service time is given by

$$CVi^2 = \frac{\sum_{r=1}^{c} (1 + CVir^2)\, Sir^2\, Vir\ Xr}{Si^2} - 1.$$

Then from the Pollaczek-Khinchin formula, the wait time for all jobs is

$$Wi = \frac{(1 + CVi^2)\, Si^2\, Xi}{2\,(1 - Xi\ Si)}.$$

The isolated system response time of class r is

$$Rir = Sir + Wi \ .$$

Transformed Model M: Because all jobs receive "non-discriminatory" service at server i in Mo, an appropriate equivalent server is a single server visited by all classes. The effective service time is computed by the forward mapping specified in Theorem 4.2.

Note that the preceding analysis assumes generally distributed service times at server i. In the special case where these service times are exponentially distributed, the model still violates product form assumptions unless all classes have the same mean service time. For exponentially distributed cases where the mean service times may differ, Bard has proposed an MVA based approximate solution [Bard 79]. Table 2 shows the accuracy of the RTP approximation for the limited set of examples considered in [Bard 79]. We also note that the RTP method provides exact results for product form queueing networks with load independent FCFS servers.

5.2 Priority Scheduling in Computer Systems:

Consider a computer system in which a device, say the CPU, gives preemptive priority to class 1 customers over class 2. The assumptions for product form solution are violated at the CPU, and thus an efficient approximation procedure is necessary. Some approximations have been discussed in [Sevcik 77, Agrawal 83b, Chandy and Laksmi 83, Bryant et. al. 83]. We now present another approximation based on the general RTP approximation technique.

The idea is to replace the CPU by equivalent CPUs, CPU1 and CPU2. The service times at these devices are computed such that the class 1 and class 2 response times at CPU1 and CPU2 in isolation are the same as the class 1 and class 2 response times at original CPU in isolation. The response times at the original CPU are obtained via auxiliary model M1.

Model M1: This model is constructed by taking the CPU out of the system and examining it in isolation. The interarrival times at the isolated CPU are assumed to be distributed exponentially. The response times can be directly calculated using well-known formulae [Kleinrock 76]:

$$R1 = S1 + \frac{(1 + CV1^2)\, S1\, U1}{2\,(1 - U1)}$$

$$R2 = \frac{S2\,(1 - U1) + ((1 + CV1^2)\, S1\, U1 + (1 + CV2^2)\, S2\, U2)/2}{(1 - U1)\,(1 - U1 - U2)},$$

where Sr is per visit CPU service time, Ur is CPU utilization and CVr is coefficient of variation for service time for class r.

Transformed Model M: In this model, the CPU is replaced by CPU1 and CPU2. The effective service times of these "shadow" CPUs are computed by using Theorem 4.1, Corollary 1.

Since the throughputs are not known initially, they are computed iteratively.

We present some numerical results in Table 3. The network under consideration is a two-station cyclic network. Two station cyclic networks are perhaps the worst-case for this algorithm because the principal source of the error in the approximation is the mismatch between the arrival process assumed for the isolated CPU analysis and the arrival process encountered at the CPU in the network.

To obtain an idea of the relative accuracy of some of the approximation methods cited earlier, we compare the errors for model 1 in note 5 of Table 3. Note that the RTP approximation, which is based on general principles that are completely independent of this particular application, compares favorably with other approximations that were specifically motivated by and tailored for the analysis of networks with preemptive priority servers. Of course, only a few cases are presented in our table, so no general conclusion can be drawn. Nevertheless, the combination of generality, simplicity, and relative accuracy exhibited by the RTP approach in this example is noteworthy.

The approximation procedure for modeling non-preemptive priority scheduling is similar to the procedure outlined above for preemptive priority. Instead of the preemptive priority equations, use non-preemptive priority equations for analysis of the open system (model M1) (e.g., equation 3.30 in [Kleinrock 76]). The accuracy of the non-preemptive priority approximation is expected to be comparable to that for the preemptive priority approximation.

5.3 Distributed Systems:

A model of a distributed processing system must represent both nodes and a communication network. The protocols used to manage the network make it difficult to treat them adequately as product form servers. However, many networks have been analyzed in isolation under Poisson arrival assumption.

The RTP approach is well suited for integrating these open-model solutions into a comprehensive product form model that represents both nodes and networks To apply the RTP approach in such cases, represent the network as a product form server. The service time for this server is obtained from existing analyses of the network operating in isolation under Poisson or homogeneous arrivals [Berry and Chandy 83, Gelenbe and Mitrani 82, Kuehn 79, Marathe and Kumar 81].

The solution procedure follows the same steps as the FCFS example given at the start of this paper, except that the Pollaczek-Khinchin formula is replaced by the appropriate equation for the network response time.

Another issue that arises when modeling distributed processing systems is the synchronization delays that occur when computations proceeding in parallel on several nodes need to coordinate their operations. In such cases, it is sometimes possible to compute the expected synchronization delays by modeling each node and network as an open system, and then adding up individual response times to determine the length of each parallel path. Once the expected delay due to path synchronization is determined, an additional server can be added to the model to represent this delay. RTP techniques can be used to determine the service time of this server.

6. Some Considerations in Developing an RTP Approximation

In this section we discuss three important considerations in developing an effective RTP based approximation. These are: the number of equivalent servers, the type of equivalent servers, and the nature of the arrival process used in the analysis of the isolated subsystems.

6.1 Number of Equivalent Servers:

This issue arises when considering a multi-class subsystem. Corollary 1 provides a forward mapping that creates several FCFS (PS) servers, each dedicated to a single class, while Theorem 4.2 can be used to create a single FCFS (PS) server capable of serving all classes. The problem is deciding which type of mapping to use when solving a specific problem.

To illustrate the issue, recall the preemptive priority scheduling system discussed in Section 5. Class 1 customers have priority over class 2 customers at the CPU and therefore they do not have to wait for class 2 customers. In this case, since class 1 customers do not suffer any contention at all from class 2 customers, we chose to use one equivalent server for each class. However, if we use only one equivalent processor sharing server for both classes, the number of low priority class 2 customers at that server would influence the completion rate of class 1 customers. As shown by the data in Table 4, the performance measures are quite inaccurate in this case. Results of intermediate calculations are also presented in Table 4. They show that some serious numerical difficulties may arise as well.

On the other hand, consider a 2-class network with different per visit service times for the two classes at an FCFS server. In this case, the two classes freely contend with each other so one multiclass equivalent server is the appropriate choice. This reasoning was used implicitly in Section 5.1.

If we use separate servers for each class, the network would be partitioned into two subnetworks, one for each class. This partitioning eliminates the dynamic interaction between the two classes that occurs in the original multiclass system. As a result both the accuracy and the numerical properties of the method suffer. This point is illustrated in Appendix B.

While the choice between a single server with multiple workloads or multiple servers with dedicated workloads is clear for these two examples, in general the decision may not be as straightforward. Fortunately the problem is not as severe as it may seem at first because an improper choice of the structure often leads to easily identifiable problems such as numerical instability. These problems serve as the indicators of inappropriate structural decisions. The analyst should be aware of these problems and experiment with different alternatives to reach a judicious conclusion.

6.2 Equivalent Server Type:

Possible types of equivalent servers are numerous and include the FCFS/PS server, the delay server, the mult-server and the load-dependent server. One aspect affecting the choice of server type is the level of concurrency in the subsystem. If the level of concurrency is low, (e.g., as in the general FCFS server examples discussed in Sections 2 and 5), an FCFS/PS server is an appropriate choice. On the other hand, if the level of concurrency in the subsystem is high, then a mult-server, a load dependent server, or even a delay server may be more appropriate. Examples of highly concurrent subsystems include mult-CPUs with priority scheduling, or a computer network with alternate paths.

A second consideration that affects the choice of server type is the complexity of the forward mapping F for calculating necessary parameters of the equivalent servers (equation (8)). When the equivalent server is a single server or a delay server, the parameter calculation is simple and the required formulae were given in Section 4. On the other hand, if the equivalent server is a mult-server with M individual processors, the forward mapping involves finding the root of an M-degree polynomial.

The third major consideration in selecting an appropriate server type is the arrival process assumed at the isolated subsystem. We discuss this issue in detail next.

6.3 Arrival Process at the Isolated Subsystem:

The arrival process assumed at the isolated subsystem not only affects the solvability of the isolated subsystem and the equivalent server, but also guides the selection of the appropriate equivalent server. The principal factor is the similarity between the arrival process observed at the subsystem when it is embedded in the original model and the one assumed at the isolated subsystem. If the two are similar, even a delay server may be adequate to represent the subsystem in the transformed approximate model. An example of such a method is Zahorjan and Lazowska's approximate MVA algorithm for networks incorporating load-dependent servers [Zahorjan and Lazowska 84]. In this algorithm, a load dependent server is replaced by a delay server. The delay is computed by analyzing the load dependent server under a load dependent arrival process generated by an equivalent server for the rest of the network. The service rates of the equivalent server for the rest of the network are computed approximately. An FCFS server can also be used, but determining its service time is more involved.

If the arrival processes in the isolated subsystem and the original model are quite different, the server type should be chosen such that the effect of the discrepancy can be mitigated. The general FCFS server example illustrates. In the original network, the interarrival times at the general server are a function of the number of customers at that server. For the isolated subsystem analysis, however, we assume that the inter-arrival times are exponentially distributed with fixed mean 1/X. As a result, the queue length at the isolated general server can exceed the number of customers in the network and the response time R can be much larger than what will be observed in the original network.

Let us now determine the type of the replacement server. The response time at a delay server is not affected by the arrival process, and therefore if a delay server is used as a replacement for the general server, the network throughput will be underestimated. The response time at an exponential FCFS server, on the other hand, depends on the arrival process in a way similar to the response time at a general FCFS server. Therefore, using an exponential FCFS server as a replacement for the general server mitigates the error due to the arrival process discrepancy.

The interaction between the arrival process and type of equivalent server also provides a clue to the accuracy of the EPF method [Shum and Buzen 77] and Marie's method [Marie 78] for solving networks containing general servers. Both of these methods accurately represent the load dependent nature of the arrival process for the general server analysis. Due to the interaction between the arrival process and service process, the response time and queue length distribution at the general server are similar to the ones that may be observed at a load dependent server. Both methods incorporate this effect. Because both methods introduce very small in each step of the approximation, their accuracy is very good.

7. RTP and Decomposition

As mentioned earlier, an RTP based approximation is essentially a decomposition approximation [Courtois 77]: we isolate a subsystem, analyze it under an "arbitrarily" assumed arrival process and use the isolated system's response time to parameterize its equivalent server(s). This technique of equating a subsystem's On-line behavior with its Off-line behavior can be regarded as a dual of Norton's Theorem approach [Chandy et. al. 75]. In the latter approach, we isolate a subsystem, analyze it as a closed system (i.e., under constant load or finite population), and use its throughput to parameterize an equivalent server. To see the duality, note that in the analysis of an open system, the throughput is usually known and the response time (mean queue length) is calculated. On the other hand, in the analysis of a closed system, the customer population (system queue length) is known and the throughput is normally calculated.

Both RTP and the Norton's Theorem approach lead to a state space transformation. The Norton's Theorem usually aggregates a set of states into a composite state and reduces the size of state space. Besides reducing the state space size by aggregation, RTP can also change the inherent state space structure and introduce new servers. This transformation is evident in the priority server analysis.

It is important to point out that these two techniques are complimentary and are not substitutes for one another. They can be effectively combined to develop solutions for complex systems. Consider, for example, a multiclass interactive system with a given maximum level of multiprogramming and priority scheduling at the CPU. To analyze such a system, we first apply the RTP approximation to obtain the central system throughputs under constant loads. Then as a second step, using the Norton's Theorem approach, we use these throughputs to characterize the equivalent server for the central system and solve the terminal-central system model. Another example of a technique that combines both approaches is the one outlined earlier for distributed systems.

Another point to be considered is when to use either RTP or the Norton's Theorem approach. The choice is usually fairly clear as their application domains are different. The Norton's Theorem approach is usually applied when the isolated subsystem's throughput can be easily computed under constant population. In its typical application, the subsystem consists of multiple devices and has a product form solution, but there is a delay in a passive resource queue before entering the subsystem.

The RTP approach, on the other hand, is applicable whenever the isolated system's response time can be easily computed under a chosen arrival process. In a typical application, the isolated subsystem violates product form assumptions, but it not preceded by a passive resource queue. Examples of such systems include FCFS general servers, priority queues, computer networks, etc.

8. Conclusion

The Response Time Preservation (RTP) is a general technique for developing approximate analysis procedures for queueing networks that contain subsystems which can be analyzed in isolation. The technique involves replacing the subsystem by equivalent servers. These servers are parameterized by using performance metrics obtained from isolated subsystems. Typically, the isolated system is analyzed as an open system, though other kinds of arrival processes, in principle, can also be used.

The RTP methodology provides an elegant, effective and efficient procedure for developing approximations. It is basically a decomposition approximation and can be regarded as a dual of the Norton's Theorem approach. The key elements of the approach entail selection of the number and type of equivalent servers as well as the arrival process used for isolated subsystem analysis. We provide practical guidelines on these matters. All these issues and the generality of the method are illustrated by presenting approximations for analyzing FCFS servers with general service times, FCFS servers in multi-class networks with different per visit service times for different classes, and priority queueing.

ACKNOWLEDGEMENTS

We are grateful to Ethan Bolker for pointing out the simple solution given by equation (5) for the linear system described by equation (6).

REFERENCES

[Agrawal 83a]
S.C. Agrawal and J.P. Buzen, "The aggregate server method for analyzing serialization delays in computer systems," ACM TOCS, Vol. 1, No. 2 (May 1983), pp. 116-143.

[Agrawal 83b]
S.C. Agrawal, "Metamodeling: A study of approximations in queueing networks," Ph.D. Dissertation, Dept. of Computer Science, Purdue University, West Lafayette, IN. (August 1983).

[Balbo 79]
G. Balbo, "Approximate solutions of queueing network models of computer systems," Ph.D. Dissertation, Dept. of Computer Science, Purdue University, West Lafayette, IN. (December 1979).

[Bard 79]
Y. Bard, "Some extensions to multiclass queueing network analysis," in Performance of Computer Systems, Arato et al. Eds., North-Holland, New York, NY, (1979) pp. 51-62.

[Bard 80]
Y. Bard, "A model of shared DASD and multipathing," Comm. ACM 23, 10 (October 1980) pp. 564-583.

[Baskett et. al. 75]
F. Baskett, K.M. Chandy, R.R. Muntz and F.G. Palacios, "Open, closed, and mixed networks of queues with different classes of customers," J. ACM 22,2 (1975) pp. 248-260.

[Berry and Chandy 83]
R. Berry and K.M. Chandy, "Performance models of token ring local area networks," Proc. 1983 ACM SIGMETRICS Conf. on Measurement and Modeling of Computer Systems, Minneapolis, August 29-31, 1983 printed as Performance Evaluation Review Special Issue August 1983, pp. 266-274.

[Brandwajn 74]
A.E. Brandwajn, "A model of a time-sharing virtual memory system solved using equivalence and decomposition methods," Acta Informatica 4,1 (1974) pp. 11-47.

[Brandwajn 82]
A.E. Brandwajn, "Fast approximate solution of multiprogramming models," Proc. 1982 ACM SIGMETRICS Conf. on Measurement and Modeling of Computer Systems, Seattle, Aug 30-Sept 1, 1982 printed as Performance Evaluation Review 11,4 (Winter 1982-83) pp. 141-149.

[Bruell and Balbo 80]
S.C. Bruell and G. Balbo, "Computational algorithms for single and multiple class closed queueing network models," Series on Programming and Operating Systems, Elsevier/North Holland Publishing Co., New York (1980).

[Bryant et. al. 83]
R.M. Bryant, A.E. Krzesinski and P. Teunissen, "The MVA pre-empt resume priority Approximation," Proc. 1983 ACM SIGMETRICS Conf. on Measurement and Modeling of Computer Systems, Minneapolis, August 29-31, 1983 printed as Performance Evaluation Review Special Issue August 1983, pp. 12-27.

[Buzen 73]
J.P. Buzen, "Computational algorithms for closed queueing networks with exponential servers," Comm. ACM 16,9 (Sept. 1973) pp. 527-531.

[Buzen and Agrawal 83]
J.P. Buzen and S.C. Agrawal, "State space transformations in queueing network models," Proc. 1983 ACM SIGMETRICS Conf. on Measurement and Modeling of Computer Systems, Minneapolis, August 29-31, 1983 printed as Performance Evaluation Review Special Issue August 1983, pp. 55-69.

[Chandy et.al. 75]
K.M. Chandy, U. Herzog and L. Woo, "Parametric analysis of queueing networks," IBM Journal of Research and Development Vol. 19, No. 1 (January 1975) pp. 36-42.

[Chandy and Sauer 78]
K.M. Chandy and C.H. Sauer, "Approximate methods for analyzing queueing networks," Computing Surveys, Vol. 10, No. 3 (September 1978) pp. 281-317.

[Chandy and Laksmi 83]
K.M. Chandy and M.S. Laksmi, "An approximation technique for queueing networks with preemptive priority queues," Technical Report Dept. of Comp. Sc., Univ. of Texas at Austin, Austin, TX (1983).

[Courtois 75]
P.J. Courtois, "Decomposability, instabilities, and saturation in multiprogramming systems," Comm. ACM 18,7 (July 1975) pp. 371-376.

[Courtois 77]
P.J. Courtois, Decomposability: Queueing and Computer System Applications, Academic Press, New York (1977).

[Denning and Buzen 78]
P.J. Denning and J.P. Buzen, "The operational analysis of queueing network models," Computing Surveys 10,3 (September 1978) pp. 225-261.

[Gelenbe and Mitrani 82]
E. Gelenbe and I. Mitrani, "Control policies in CSMA local area networks: Ethernet controls," Proc. 1982 ACM SIGMETRICS Conf. on Measurement and Modeling of Computer Systems, Seattle, Aug 30-Sept 1, 1982 printed as Performance Evaluation Review 11,4 (Winter 1982-83) pp. 233-240.

[Graham 78]
G.S. Graham, Editor, "Special issue on Queueing Network Models," Computing Surveys, 10,3 (Sept. 1978).

[Kleinrock 75]
L. Kleinrock, Queueing Systems Volume I: Theory, Wiley, New York, NY (1975).

[Kleinrock 76]
L. Kleinrock, Queueing Systems Volume II: Computer Applications, Wiley, New York, NY (1976).

[Kobayashi 78]
H. Kobayashi, Modeling and Analysis: An Introduction to System Performance Evaluation Methodology, Addison-Wesley Publishing Co., Reading, Mass. (1978).

[Kuehn 79]
Kuehn, P.J., "Multiqueue Systems with Nonexhaustive Cyclic Service," BTSJ, Vol. 58(3) pp. 671-698 (1979).

[Lazowska and Zahorjan 82]
E.D., Lazowska and J. Zahorjan, "Multiple class memory constrained queueing networks," Proc. 1982 ACM SIGMETRICS Conf. on Measurement and Modeling of Computer Systems, Seattle, Aug 30-Sept 1, 1982 printed as Performance Evaluation Review 11,4 (Winter 1982-83) pp. 130-140.

[Marathe and Kumar 81]
M. Marathe and S. Kumar, "Analytical models for an Ethernet-like local area network," Proc. 1981 ACM SIGMETRICS Conf. on Measurement and Modeling of Computer Systems, Las Vegas, September 14-16, 1981 printed as Performance Evaluation Review 10,3 (Fall 1981-82) pp. 205-215.

[Marie 78]
R. Marie, "Modelisation par resseaux de files d'attente," Ph.D. Thesis Universite' de Rennes, France, (December 1978).

[Potier and LeBlanc 80]
D. Potier and Ph. LeBlanc, "Analysis of locking policies in database management systems," Comm. ACM 23,10 (October 1980) pp. 584-593.

[Reiser and Lavenberg 80]
M. Reiser and S.S. Lavenberg, "Mean value analysis of closed multichain queueing networks," J. ACM 27,2 (April 1980) pp./ 3113-322.

[Sevcik 77]
K.C. Sevcik, "Priority scheduling disciplines in queueing network models of computer systems" Proc. IFIP Congress 77, North-Holland Publishing Co., Amsterdam (1977) pp. 565-570.

[Shum and Buzen 77]
A.W. Shum and J.P. Buzen, "The EPF technique: A method for obtaining approximate solutions to closed queueing networks with general service times," Proc. Third Symposium on Measuring Modeling, and Evaluating Computer Systems, Bonn-Bad Godesborg, B.R.D., North Holland (1977) pp. 201-220.

[Zahorjan and Lazowska 84]
J. Zahorjan and E. Lazowska, "Incorporating load dependent servers in approximate mean value analysis", University of Washington Tech. Rpt. 84-02-01, (February 1984).

Appendix A

ACCURACY OF RTP APPROXIMATION FOR FCFS SERVERS

We now consider the results of a systematic study of a machine repairman model and a two FCFS queue cyclic network. These two systems represent two extreme cases. In the machine repairman model the general server is subjected to a load dependent arrival process, with arrival rate

$$Am(n) = \begin{cases} \dfrac{N-n}{\text{think time}} = (N-n)\,A, & n=0,\ldots,N-1 \\ 0 & n>N. \end{cases}$$

In the two queue models, the general server is subjected to a fixed rate arrival process:

$$At(n) = \begin{cases} M1 & n = 0, \ldots, N-1 \\ 0 & n>N, \end{cases}$$

where M1 = 1/S1 is the service rate of the server 1. In a real system, the arrival process at the general server will have some intermediate arrival rates

$$Ar < Ar(n) < (N-n)Ar$$

(and possibly $Ar(n-1) < Ar(n)$). Therefore, the evaluation of the accuracy of the RTP approximation for these two models can provide a good indication of the method's accuracy.

The machine repairman model is solved exactly as an M/G/1//N system [Buzen and Goldberg 74]. Table 5 presents the results of the study. In the experiment, the mean and the coefficient variation of the service time were 1.0 and CV, respectively. THINK is the think time, N is the number of customers in the network; RN is the response time of the general server and XN is the network throughput. (XN also equals the general server utilization). ERA and EXA are the relative percent errors in the RTP estimates of general server response time and system throughput, respectively. ERP and EXP are the relative percent errors in the corresponding estimates computed by ignoring the coefficient of variation, i.e., by assuming that the model has a product form solution. Some important observations follow. The throughput estimates are quite accurate even at high CV's and moderate number of terminals (> 5). The errors in the device response times are much larger, especially at large CV's (> 5) but decrease to tolerable levels at 5 or more terminals. Maximum errors occur when the general server utilization is about 50%. A comparison with product form solution shows that RTP approximation substantially increases accuracy when a number of terminals is 5 or more.

The two queue cyclic model can be solved exactly as an M/G/1/N loss system [Kobayashi 78]. Table 6 presents the results of the study. In the table, the new variable S1 is the service time of the exponential server. Once again, we see that the RTP approximation is a fairly effective technique, especially for moderate to large numbers of customers (> 5) and low to moderate CV's (< 5).

Appendix B

EFFECT OF NUMBER OF EQUIVALENT SERVERS

Section 6.1 addressed the issue of choosing the appropriate number of equivalent servers to represent a subnetwork Mi in the transformed model M. If the customer classes in Mi contend freely with each other (as in the multiclass FCFS example in Section 5.1), using a separate server for each will eliminate the dynamic interaction present in the original system. Both the accuracy and the numerical properties of the RTP approximation will then suffer.

One indication of the deterioration of accuracy is that the RTP approximation no longer yields the exact solution when it is applied to a multiclass product form network consisting only of load independent FCFS servers. The data in Table 7a illustrates this. Another indication is provided by the data presented in Table 7b. For the network under consideration, note that class 2 service time at device 1 is less than that of class 1; at device 2, they are comparable; and at device 3, which is very lightly used, class 2 has higher demand than class 1. Therefore, we expect that for equal class populations, class 2's response time should be smaller than class 1's response time. The solutions from both simulation and RTPA1 tally with this observation. But for smaller populations, RTPA2 does not. Partitioning the subnetwork by customer class also creates certain numerical difficulties. Convergence with separate equivalent servers is painfully slow (100-300 iterations). Moreover, due to incorrect intermediate values of the throughputs, servers can easily become saturated during the iteration.

There are two solutions to the saturation problem. The first one assumes that since the server is saturated, the response time is infinite, and thus, equation (3) reduces to

$$Sir' = 1 / Xir.$$

With this assumption, however, the solution diverges.

The second solution to the saturation problem reduces the throughput estimates in proportion to the individual class utilizations such that the total server utilization is UMAX. This method led to a slow convergence with UMAX = 0.99. However, the iteration diverged with UMAX = 0.9999.

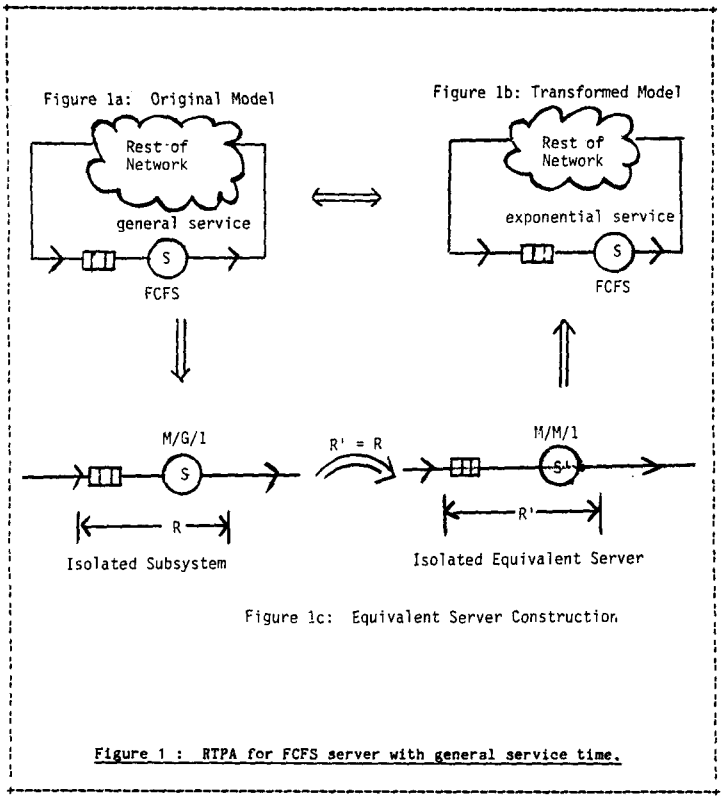

Figure 1c: Equivalent Server Construction

Figure 1 : RTPA for FCFS server with general service time.

Table 1 : General service time at FCFS server example.

i	Vi	Si	ViSi
CPU	10	28 ms	280 ms
DISK 1	7	40 ms	280 ms
DISK 2	2	280 ms	560 ms

NETWORK POPULATION N = 6

MODEL #	CPU	CV DISK 1	CV DISK 2	NO.OF ITER.	CPU THROUGHPUT EXACT	CPU THROUGHPUT APPROX.	% ERROR
1	0.6	1.0	1.0	1	0.01744	0.01743	- 0.1
2	2.0	1.0	1.0	1	0.01703	0.01695	- 0.5
3	5.0	1.0	1.0	4	0.01588	0.01517	- 4.5
4	10.0	1.0	1.0	8	0.01487	0.01310	- 11.9
5	1.0	0.6	1.0	1	0.01745	0.01743	- 0.1
6	1.0	2.0	1.0	1	0.01703	0.01695	- 0.5
7	1.0	10.0	1.0	8	0.01483	0.01310	- 11.7
8	1.0	1.0	0.6	3	0.01765	0.01750	- 0.8
9	1.0	1.0	2.0	4	0.01662	0.01687	1.5
10	1.0	1.0	5.0	17	0.01567	0.01531	- 2.3
11	1.0	1.0	10.0	34	0.01541	0.01368	- 11.2
12	0.6	2.0	10.0	31	0.01500	0.01278	- 14.8
13	10.0	2.0	0.6	8	0.01489	0.01328	- 10.8
14	5.0	2.0	10.0	21	0.01363	0.00942	- 30.9
15	10.0	2.0	0.6	33	0.01353	0.00886	- 34.5

NOTES: These are some of the models evaluated in [Balbo 79].
The exact solutions were obtained by Balbo using global balance techniques.

If the CV's are ignored,
CPU throughput for all models = 0.01735.

We are unable to explain the error for model 9.

Table 2 : Multi-class FCFS server example.

DEVICE i	CLASS 1 Vi1	Si1	Vi1Si1	CLASS 2 Vi2	Si2	Vi2Si2
1	18	7	126	8	11	88
2	5	20	100	13	8	104
3	10	2	20	15	3	45

POPULATION CLASS 1	POPULATION CLASS 2	# RTPA ITER.	RESPONSE TIME CLASS 1 SIM.	CLASS 1 RTPA	CLASS 1 BARD	CLASS 2 SIM	CLASS 2 RTPA	CLASS 2 BARD
5	5	8	1140	1115	1154	1156	1120	1208
50	50	14	12267	12952	12022	9070	8566	9316

NOTE: The example and simulation results are taken from [Bard 79].

Table 3 : Preemptive priority examples.

MODEL #	CLASS	S CPU	S DISK	THROUGHPUT EXACT	THROUGHPUT RTPA	% ERROR
1	1	3	3	0.2156	0.2066	1.9
	2	3	3	0.0807	0.1117	13.3
2	1	3	1	0.3302	0.3305	0.1
	2	3	1	0.0031	0.0031	0
3	1	1	3	0.2566	0.2583	0.7
	2	3	1	0.2261	0.2227	- 1.5
4	1	3	1	0.3282	0.3303	0.6
	2	1	3	0.0103	0.0100	- 2.9
5	1	1	1	0.6446	0.6472	0.4
	3	3	3	0.0885	0.0977	10.4

NOTES:
1. System is a 2-queue cyclic network.
2. Customer population of each class = 4.
3. Exact throughputs were calculated using global-balance solution technique.
4. With given approximate solution for models 2 and 4, CPU utilization exceeds 1. When this happens, reduce class 1 throughput to make utilization 1. Therefore:

MODEL	ADJUSTED X1	%ERROR
2	0.3302	0
4	0.3300	0.5

5. COMPARISON OF DIFFERENT METHODS
% ERROR IN THROUGHPUT

MODEL #	CLASS	RTPA	SEVCIK	BRYANT ET. AL.	CHANDY LAKSMI
1	1	1.9	- 4.2	0.2	- 5.2
	2	13.3	38.4	- 6.1	21.9

Table 4 : Priority scheduling with one equivalent server

ITER.	ARRIVAL RATE 1	ARRIVAL RATE 2	RESP TIME 1	RESP TIME 2	EFFECTIVE SERVICE TIME 1	EFFECTIVE SERVICE TIME 2	THROUGHPUT 1	THROUGHPUT 2
1	.2156	.0807	8.5	95	.8093	9.05	.2424	.0823
2	.2424	.0823	11.0	528	.2335	11.21	.2550	.0782
3	.2550	.0782	12.8	34329	.0048	12.77	.2597	.0739
4	.2597	.0737	13.6	1.7e12	1.e-11	13.57	see note 2	

NOTES:
1. Initial guess is exact solution from global balance solution.
2. The iteration seems to be stabilizing around X1 = 0.26. and X2 = 0.07. Iteration was discontinued at this point because of too small a value for class 1 effective service time. Nonetheless, these results show that the answers are rather inaccurate.

MODEL: 2 queue cyclic network with priority queueing at server 1.
Dir = 3, Nr = 4, i = 1,2, r = 1,2.

Table 5 : Accuracy of RTP for Machine Repairman Model

CV	THINK	N	RN	ERA	ERP	XN	EXA	EXP
0.0	0.400	2	1.633	1.8	5.0	0.984	-1.4	-3.9
0.0	1.000	5	4.000	0.2	0.4	1.000	-0.2	-0.3
0.0	2.000	10	8.000	0.0	0.0	1.000	-0.0	-0.0
0.0	1.000	2	1.368	-2.6	9.7	0.845	1.5	-5.3
0.0	2.500	5	2.641	2.9	8.8	0.973	-1.5	-4.3
0.0	5.000	10	5.024	1.5	3.3	0.998	-0.8	-1.6
0.0	2.000	2	1.213	-8.4	9.9	0.622	3.3	-3.6
0.0	5.000	5	1.663	-2.8	19.8	0.750	0.7	-4.7
0.0	10.000	10	2.180	0.2	25.3	0.821	-0.0	-4.3
0.0	5.000	2	1.094	-7.2	6.7	0.328	1.3	-1.2
0.0	12.500	5	1.199	-3.9	13.9	0.365	0.3	-1.2
0.0	25.000	10	1.254	-2.3	18.0	0.381	0.1	-0.9
0.0	10.000	2	1.048	-4.4	4.1	0.181	0.4	-0.4
0.0	25.000	5	1.089	-2.1	7.6	0.192	0.1	-0.3
0.0	50.000	10	1.106	-1.1	9.2	0.196	0.0	-0.2
2.000	0.400	2	1.751	5.1	-2.1	0.930	-4.0	1.7
2.000	1.000	5	4.047	-0.2	-0.8	0.991	0.2	0.6
2.000	2.000	10	8.005	-0.1	-0.1	1.000	0.0	0.0
2.000	1.000	2	1.588	13.9	-5.6	0.773	-7.9	3.5
2.000	2.500	5	3.121	2.6	-7.9	0.890	-1.4	4.6
2.000	5.000	10	5.501	-1.8	-5.7	0.952	0.9	3.1
2.000	2.000	2	1.455	20.3	-8.3	0.579	-7.9	3.6
2.000	5.000	5	2.406	10.4	-17.2	0.675	-3.3	5.9
2.000	10.000	10	3.514	4.5	-22.2	0.740	-1.2	6.1
2.000	5.000	2	1.286	19.8	-9.3	0.318	-3.9	1.9
2.000	12.500	5	1.699	12.3	-19.7	0.352	-1.5	2.4
2.000	25.000	10	1.994	8.1	-25.8	0.370	-0.6	1.9
2.000	10.000	2	1.180	13.4	-7.6	0.179	-1.4	0.8
2.000	25.000	5	1.377	7.3	-14.9	0.190	-0.4	0.8
2.000	50.000	10	1.480	4.2	-18.4	0.194	-0.1	0.5
5.000	0.400	2	1.772	30.1	-3.3	0.921	-19.7	2.8
5.000	1.000	5	4.083	2.0	-1.7	0.984	-1.6	1.3
5.000	2.000	10	8.019	-0.2	-0.2	0.998	0.2	0.2
5.000	1.000	2	1.650	63.4	-9.1	0.755	-28.3	6.0
5.000	2.500	5	3.395	23.5	-15.3	0.848	-11.9	9.7
5.000	5.000	10	6.061	2.8	-14.4	0.904	-1.5	8.6
5.000	2.000	2	1.565	89.3	-14.8	0.561	-28.2	7.0
5.000	5.000	5	2.973	49.1	-33.0	0.627	-15.5	14.8
5.000	10.000	10	4.897	23.2	-44.2	0.671	-7.1	17.0
5.000	5.000	2	1.464	105.1	-20.3	0.309	-19.2	4.8
5.000	12.500	5	2.477	67.0	-44.9	0.334	-10.0	8.0
5.000	25.000	10	3.579	42.9	-58.7	0.350	-5.1	7.9
5.000	10.000	2	1.383	90.5	-21.1	0.176	-9.9	2.6
5.000	25.000	5	2.084	54.7	-43.8	0.185	-4.0	3.5
5.000	50.000	10	2.678	34.0	-54.9	0.190	-1.7	2.9

Table 6 : Accuracy of RTP for Two Queue Cyclic Model

CV	S1	N	RN	ERA	EXP	XN	EXA	EXP
0.0	0.200	2	1.801	-0.0	1.8	0.999	-1.6	-3.1
0.0	0.200	5	4.799	-1.0	-1.0	1.000	-0.0	-0.0
0.0	0.200	10	9.799	-0.5	-0.5	1.000	0.0	0.0
0.0	0.500	2	1.568	-2.3	6.3	0.937	-3.2	-8.5
0.0	0.500	5	4.377	-5.9	-4.9	0.999	-0.8	-1.5
0.0	0.500	10	9.372	-3.9	-3.9	1.000	-0.0	-0.0
0.0	1.000	2	1.368	-7.7	9.7	0.731	-2.3	-8.8
0.0	1.000	5	2.840	-10.0	5.6	0.897	-2.9	-7.1
0.0	1.000	10	5.335	-10.9	3.1	0.949	-1.7	-4.2
0.0	2.500	2	1.176	-6.2	9.3	0.374	-1.9	-3.9
0.0	2.500	5	1.329	-1.3	21.5	0.400	-0.3	-0.6
0.0	2.500	10	1.333	0.0	24.9	0.400	0.0	-0.0
0.0	5.000	2	1.094	-2.8	6.7	0.196	-1.0	-1.4
0.0	5.000	5	1.125	-0.1	11.0	0.200	-0.0	-0.0
0.0	5.000	10	1.125	0.0	11.1	0.200	0.0	0.0
0.707	0.200	2	1.816	0.1	0.9	0.984	-0.9	-1.6
0.707	0.200	5	4.779	-0.6	-0.6	1.000	-0.0	-0.0
0.707	0.200	10	9.779	-0.3	-0.3	1.000	0.0	0.0
0.707	0.500	2	1.625	-1.3	2.6	0.889	-1.1	-3.6
0.707	0.500	5	4.244	-2.5	-2.0	0.994	-0.7	-1.0
0.707	0.500	10	9.192	-2.0	-2.0	1.000	-0.0	-0.0
0.707	1.000	2	1.444	-3.5	3.8	0.692	-0.7	-3.7
0.707	1.000	5	2.923	-4.0	2.6	0.862	-1.4	-3.3
0.707	1.000	10	5.420	-4.4	1.5	0.928	-0.9	-2.0
0.707	2.500	2	1.236	-3.1	4.0	0.365	-0.8	-1.8
0.707	2.500	5	1.480	-1.0	9.1	0.399	-0.2	-0.4
0.707	2.500	10	1.500	-0.0	11.0	0.400	-0.0	-0.0
0.707	5.000	2	1.132	-1.5	3.0	0.195	-0.4	-0.7
0.707	5.000	5	1.187	-0.1	5.2	0.200	-0.0	-0.0
0.707	5.000	10	1.188	-0.0	5.3	0.200	0.0	0.0
2.000	0.200	2	1.847	3.7	-0.8	0.955	-2.5	1.4
2.000	0.200	5	4.720	0.7	0.7	0.999	0.1	0.1
2.000	0.200	10	9.711	0.4	0.4	1.000	0.0	0.0
2.000	0.500	2	1.714	12.4	-2.8	0.824	-5.9	4.1
2.000	0.500	5	4.060	5.2	2.5	0.955	1.1	3.0
2.000	0.500	10	8.576	5.1	5.1	0.994	0.5	0.6
2.000	1.000	2	1.588	20.9	-5.6	0.630	-5.7	5.9
2.000	1.000	5	3.154	17.4	-4.9	0.760	2.1	9.7
2.000	1.000	10	5.614	17.6	-2.0	0.839	3.8	8.4
2.000	2.500	2	1.412	22.1	-8.9	0.343	-0.9	4.5
2.000	2.500	5	2.089	16.0	-22.7	0.384	2.3	3.5
2.000	2.500	10	2.515	5.4	-33.8	0.397	0.6	0.7
2.000	5.000	2	1.286	13.5	-9.3	0.189	0.9	2.3
2.000	5.000	5	1.565	3.5	-20.2	0.199	0.5	0.6
2.000	5.000	10	1.622	0.2	-22.9	0.200	0.0	0.0
5.000	0.200	2	1.855	22.0	-1.2	0.948	-15.3	2.1
5.000	0.200	5	4.698	1.3	1.1	0.997	0.0	0.2
5.000	0.200	10	9.678	0.7	0.7	1.000	0.0	0.0
5.000	0.500	2	1.743	54.6	-4.4	0.805	-25.1	6.5
5.000	0.500	5	4.010	18.2	3.8	0.921	-3.1	6.9
5.000	0.500	10	8.008	13.1	12.5	0.969	2.7	3.1
5.000	1.000	2	1.650	86.0	-9.1	0.606	-25.3	10.0
5.000	1.000	5	3.345	58.6	-10.3	0.685	-4.2	21.7
5.000	1.000	10	5.849	51.0	-6.0	0.723	8.9	25.7
5.000	2.500	2	1.540	114.8	-16.5	0.329	-14.7	9.1
5.000	2.500	5	2.791	97.0	-42.1	0.351	2.6	13.4
5.000	2.500	10	4.391	75.3	-62.1	0.366	6.6	9.2
5.000	5.000	2	1.464	103.1	-20.3	0.183	-2.3	5.8
5.000	5.000	5	2.416	65.8	-48.3	0.191	3.8	5.0
5.000	5.000	10	3.321	27.7	-62.4	0.196	2.1	2.2

Table 7 : Problems associated with separate equivalent servers for a multiclass FCFS network.

(a) Product form network

DEVICE i	CLASS 1 Vi1	Si1	Vi1Si1	CLASS 2 Vi2	Si2	Vi2Si2
1	10	1	10	8	1	8
2	12	1	15	12	1	14
3	15	2	30	10	2	20

POPULATION CLASS 1	POPULATION CLASS 2	RESPONSE TIME CLASS 1 EXACT	RTPA1	RTPA2	CLASS 2 EXACT	RTPA1	RTPA2
5	5	291.8	291.8	277.0	207.3	207.3	216.1

RTPA1 = 1 iteration
RTPA2 = 148 iterations
RTPA1 = RTP approximation with one equivalent server for all classes
RTPA2 = RTP approximation with one equivalent server for each class

(b) Non-product form network [Bard 79]

DEVICE i	CLASS 1 Vi1	Si1	Vi1Si1	CLASS 2 Vi2	Si2	Vi2Si2
1	18	7	126	8	11	88
2	5	20	100	13	8	104
3	10	2	20	15	3	45

POPULATION CLASS 1	POPULATION CLASS 2	RESPONSE TIMES CLASS 1 SIM	RTPA1	RTPA2	CLASS 2 SIM	RTPA1	RTPA2
5	5	1140	1115	1107	1156	1120	1228
10	10	--	2255	2048	--	2016	2444
20	20	--	4827	3749	--	3683	5436
50	50	12267	12952	11588	9070	8566	9571

-- No simulation data available in [Bard 79].