



Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

W.P. Groenendijk, H. Levy

Performance analysis of transaction driven computer systems
via queueing analysis of polling models

The Centre for Mathematics and Computer Science is a research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a nonprofit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands Organization for the Advancement of Research (N.W.O.).

Performance Analysis of Transaction Driven Computer Systems Via Queueing Analysis of Polling Models

Wim P. Groenendijk¹

Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

Hanoch Levy¹

Computer Science Department, Tel-Aviv University
Tel-Aviv, Israel

The objective of this paper is the modeling and analysis of a class of computer systems whose primary task is the massive processing of batch transactions and which we call *Transaction Driven Computer Systems (TDCS's)*. We present a generic queueing model for Transaction Driven Computer Systems and we calculate the mean sojourn time experienced by the different transactions. Our approach is to model a TDCS by a cyclic polling system with bulk arrivals, deterministic service times, limited-1 service and zero switch-over periods. Since the performance analysis of this polling model has not been provided before we concentrate in the paper on deriving mean delay approximations for it. Our analysis is carried out for models with general switch-over periods, and a special case of it (zero switch-over periods) is suitable for analyzing a TDCS.

1980 Mathematics Subject Classification: 60K25, 68M20.

Key Words & Phrases: Transaction Driven Computer System, polling system, waiting-time approximation

Note: This report will be submitted for publication elsewhere.

1. Introduction

The objective of this paper is the modeling and analysis of a class of computer systems whose primary task is the massive processing of batch transactions and which we call *Transaction Driven Computer Systems (TDCS's)*. An example of a TDCS is a *gateway machine* which is used as a connection point between several computer networks. The gateway receives streams of messages from the different networks, processes each message (e.g., retrieves the destination information, translates the message header, etc.) and directs it to the proper network. Another example is a *network server machine* which is in charge of administrating the network. The network server receives streams of messages regarding the network condition (e.g. link and node up-down status, link reservations, traffic demand, etc.), processes them, updates its internal tables, and issues control instructions to the network.

A Transaction Driven Computer System usually consists of N programs residing on a single machine. Each program receives an input stream of transactions and is responsible for processing them in the order received. Each program is equipped with an input queue for storing its transactions while waiting to be processed. The users of a TDCS as well as its programmers may consider these programs as being functionally independent of each other. For example, on a Unix system such independence is usually provided for by running each program as a separate process. Nonetheless, the performance of each program *does depend* on the behavior of the other programs, since all of them share the same operating system and the same processor.

A major issue dealt with in the design of a TDCS is the proper sharing of the processor (CPU) by the different programs. The fact that the processor is shared among the programs has two implications: 1) The design and behavior of one program affect the performance of the other programs, and 2) A proper allocation of the processor to the different programs can be used to prioritize the

1. Part of this research was done while the authors were with AT&T Bell Laboratories.

different transactions and to improve system performance. Our focus is, therefore, on analyzing the system performance as observed by the different transactions as a function of their CPU requirement and other parameters.

A model of a TDCS (consisting of N programs) which captures these features is depicted in Figure 1.1. The i th program receives a Poisson stream of type- i transactions whose rate is λ_i and whose processor time requirement (measured in processor time units) is a discrete time random variable X_i with first two moments x_i and $x_i^{(2)}$. The processor is shared among the programs in the following way: 1) Programs which do not have any transaction to be processed are "sleeping" and do not receive any processing. 2) Programs which have at least one transaction to be processed share the processor in a Round-Robin fashion. This means that the processor cycles among these programs and grants one time unit of processing at every visit to a program. 3) A transaction leaves the system once it has received the number of processing units it requires. The transactions appearing in the figure are either at the head of their queue and being processed (the dashed boxes in the figure) or waiting to be processed. The size of each transaction in the figure represents its residual processing time.

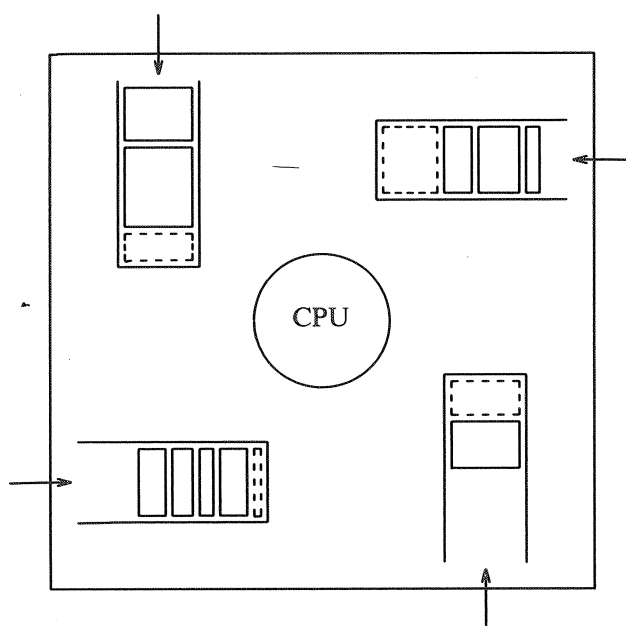


Figure 1.1 Model of a Transaction Driven Computer System (TDCS)

Our objective in this paper is to present a generic queueing model for Transaction Driven Computer Systems and to calculate the mean sojourn time experienced by the different transactions. Our approach is to model a TDCS by a cyclic polling system with bulk arrivals, deterministic service times, limited-1 service and zero switch-over periods. Since the analysis of this polling model has not been provided before we concentrate in the paper on deriving mean delay approximations for it. Our analysis is carried out for models with general switch-over periods, and a special case of it (zero switch-over periods) is suitable for analyzing a TDCS. An alternative approach for modeling a TDCS has been provided in an independent study by Leung [1988]; see details in Section 1.1.

A cyclic polling system with bulk arrivals and limited-1 service consists of N queues (indexed 1, 2, ..., N) and a single server which visits them in a cyclic order. Customers are assumed to arrive at queue i as an independent Poisson stream with rate λ_i . Each arrival to queue i consists of a bulk of K_i customers, which is a generally distributed random variable with mean k_i and second moment $k_i^{(2)}$. The service time of a type- i customer is a generally distributed random variable B_i with mean b_i and

second moment $b_i^{(2)}$. As stated earlier, the server visits the queues in cyclic order; therefore, the reference to queue indices is done modulo N . The service discipline is the *limited-1 service* in which during each visit at queue i the server will serve exactly one customer (if any is present at the queue).¹ After visiting queue i , the server switches to queue $i + 1$. The period during which the server switches from queue i to queue $i + 1$ is called a *type- i switch-over period*. The duration of this period is assumed to be a generally distributed random variable S_i with mean s_i and second moment $s_i^{(2)}$. It is assumed that the arrival, service and switch-over processes are mutually independent.

The representation of a TDCS by a cyclic polling system is straight-forward. A transaction whose processing time is X_i time units is represented by a bulk consisting of $K_i (= X_i)$ customers, with deterministic service time of one time unit (namely, $b_i = b_i^{(2)} = 1$). The sojourn time of the transaction is represented by the sojourn time of the last customer in the bulk in the polling system. In addition, it is assumed that the switch-over periods in the cyclic polling model are of zero length. As an example consider Figure 1.2 which depicts the cyclic polling model that we intend to use for the performance analysis of a TDCS as in Figure 1.1. Our approximate analysis is provided for polling systems with general switch-over periods and general service times. In the numerical investigation, however, we concentrate on examining systems which represent TDCS's, namely with zero switch-over period and deterministic (equal to 1) service times.

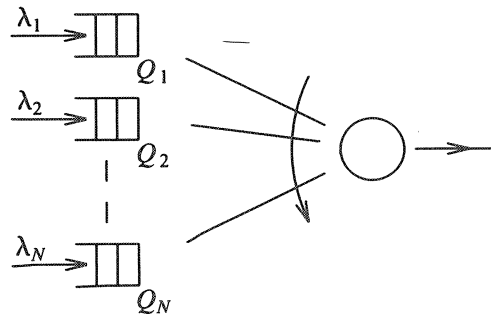


Figure 1.2 Queueing model of a TDCS

This paper is organized as follows: In Section 2 we provide a method for approximating the mean delay in the polling model. In Section 3 we conduct an extensive examination of the approximation by comparing its predictions to simulation results. The method is shown to yield good approximations for a wide range of parameters, and fails mainly in cases which combine high load with high degree of asymmetry. In Section 4 we describe a complementary procedure which is suitable for approximating the cases in which the first method fails. According to this procedure a highly utilized queue is replaced by an "equivalent" switch-over period. The numerical results of this procedure are discussed in this section as well. Concluding remarks, extensions and applications are discussed in Section 5. Finally, the appendix contains the details of the numerical results.

1.1. Previous Work

Cyclic polling systems with limited service have received considerable attention in recent years. Unfortunately, these systems in general seem to be too complex to be analyzed exactly and therefore research in this area concentrated on either solving special cases (e.g., fully symmetric systems) or deriving general approximations.

Kühn [1979] provided a mean delay approximation for cyclic polling systems with limited-1 service.

1. A generalization of the limited-1 discipline is the *limited- L_i* discipline, in which up to L_i customers are served during each visit at queue i .

His approximation is provided for systems with single customer arrivals as well as bulk arrivals.

An important contribution to the area of polling systems has been the introduction of a (pseudo) conservation law. This law, which was derived in its general form by Boxma and Groenendijk [1987] (and earlier, in less general forms, by Ferguson and Aminetazh [1985] and Watson [1985]), provides a closed-form expression for the weighted sum of the mean waiting times, $\sum_{i=1}^N \rho_i E[W_i]$ (where ρ_i is the utilization of queue i and W_i is the waiting time of a customer in that queue), in cyclic polling systems. Boxma and Meister [1986,1987] used several of the ideas presented in Kühn [1979] to derive an approximation for the ratio $E[W_i]/E[W_j]$. They combined these ratios with the conservation law to obtain an approximation of the mean waiting times in a cyclic polling system with limited-1 service and single arrivals.

Following Boxma and Meister [1986], Fuhrmann and Wang [1988] suggested a different approximation which is also based on the conservation law. Numerical examples seem to indicate that the Fuhrmann-Wang approximation works better at high loads while the Boxma-Meister approximation is better at low to medium loads. The Fuhrmann-Wang approximation applies, in addition, to more general limited-service systems in which the server serves at most L_i customers in each visit at queue i . Both approximations were derived for systems with single arrivals.

The conservation law has recently been extended and applied to systems with correlated arrivals by Boxma [1989] and by Levy and Sidi [1988]. As a special case, the extended law applies to systems with bulk arrivals. This generalized law allows the extension of the previous approximations and their application to systems with limited service in combination with bulk arrivals.

To the best of our knowledge systems like TDCS have not been analyzed previously. In an independent study Leung [1988] supplies an approximate analysis for a Processor-Sharing Background-Job system which can serve as an alternative model for TDCS. The model considered in that work consists of N "background job" queues and one "interactive job" queue. Each of the queues receives its own Poisson stream of arrivals. The service policy is a special processor sharing regime in which background jobs residing in the head of their queue and *all* the interactive jobs concurrently share the processor in a processor sharing mode. Leung [1988] supplies an approximate analysis of this system under the assumption that all service times are exponentially distributed.

The common denominator of the two models is the service of the N queues: The processor is shared among all the transactions which reside in the heads of these queues. In our model the sharing is done in a Round-Robin fashion while in Leung's model it is done in a Processor-Sharing fashion. Note that if we disregard the interactive-job queue, the two models become identical when the time unit (of the Round-Robin regime) shrinks to zero.

The main differences between the two models are as follows:

- 1) The interactive jobs, which are represented in Leung's model, are not captured in the polling model described above. However, it is easy to accommodate such jobs in our model by adding a large number (say M) of queues to store these jobs, and by having the arrival rate of each of these queues equal to $1/M$ of interactive job total arrival rate. The selection of a large M should provide that most of the interactive jobs will not have to queue up and will share the processor as they join the system.
- 2) Our model allows for general distribution of transaction service times while Leung's model allows only for exponentially distributed service times.

2. Analysis

Our goal in this section is to derive expressions for the mean waiting time of an *arbitrary customer* at queue i ($E[W_i]$) and the mean waiting time of the *last customer in a bulk* at queue i ($E[W_i^L]$). In addition to the notation introduced in Section 1, let the utilization of queue i be $\rho_i = \lambda_i k_i b_i$, and the system utilization be $\rho = \sum_{i=1}^N \rho_i$. Also, let s and $s^{(2)}$ be the first two moments of the sum of the switch-over times ($\sum_{i=1}^N S_i$); namely, $s = \sum_{i=1}^N s_i$ and $s^{(2)} = s^2 + \sum_{i=1}^N (s_i^{(2)} - s_i^2)$. Let $E[W_i]$ be the waiting time (time from joining the system until getting into service) of an arbitrary customer at queue i .

Throughout the paper the system is assumed to be in steady state.

To conduct the analysis, several additional variables need be defined. An *i-cycle* is the period between two consecutive visits to queue *i* (its boundary points are the polling instants). Let C_i be the duration of an *i-cycle* and R_i be its forward recurrence time (residual life). It is well known that the mean length of an *i-cycle*, $E[C_i]$, is independent of the queue index and is given by

$$E[C_i] = E[C] = s/(1-\rho).$$

A *busy i-cycle* is an *i-cycle* in which queue *i* is served (i.e., the queue is non-empty at the beginning of the cycle). Let $C_{b,i}$ be the duration of a busy *i-cycle*. Unlike the cycle time, $E[C_{b,i}]$ is not necessarily identical to $E[C_{b,j}]$ ($j \neq i$), and an expression for its value is not generally known.

Let us consider a tagged customer and calculate the expected number of customers arriving together with the tagged customer to the system (we call them *bulk-mates*) but which are ahead of him in the bulk. Conditioning on the size of the bulk (k) in which the customer arrives, this number is given by $(k-1)/2$. Unconditioning (noticing that the probability that an arbitrary customer of queue *i* belongs to a bulk of size k is proportional to k) yields:

$$\begin{aligned} E[\text{ \# of bulk-mates ahead of a tagged-customer }] &= \frac{\sum_{k=1}^{\infty} Pr[K_i = k] \cdot k(k-1)/2}{\sum_{k=1}^{\infty} Pr[K_i = k] \cdot k} \\ &= \frac{E[K_i^2 - K_i]}{2E[K_i]} = \frac{k_i^{(2)} - k_i}{2k_i}. \end{aligned} \quad (2.1)$$

Let X_i be the number of customers present at queue *i* at an arbitrary moment. Due to the PASTA (Poisson Arrivals See Time Averages, cf. Wolff [1982]) property, this is also the number of customers found at queue *i* by an arriving bulk.

The waiting time of an arbitrary customer at queue *i* consists of three components:

- 1) The time from the customer arrival until the next visit of the server to queue *i*; this is equal to R_i .
- 2) The time required for the server to serve the X_i customers found at the queue by the arriving customer; the duration of this period is the sum of the lengths of X_i busy *i-cycles*.
- 3) The time required by the server to serve the customers which are ahead of the tagged customer in the bulk.

Thus the mean waiting time is given by:

$$E[W_i] = E[R_i] + \left[E[X_i] + \frac{k_i^{(2)} - k_i}{2k_i} \right] E[C_{b,i}]. \quad (2.2)$$

In addition, from Little's law we have:

$$E[X_i] = k_i \lambda_i E[W_i]. \quad (2.3)$$

The value of $E[C_{b,i}]$ is approximately determined by solving the following set of equations (cf. Srinivasan [1988], Groenendijk [1989]):

$$E[C_{b,i}] \approx b_i + \sum_{j \neq i} \min(1, \lambda_j k_j E[C_{b,i}]) \beta_j + s, \quad (2.4)$$

which can be solved using a simple iteration. To evaluate $E[R_i]$ we use the approximation suggested by Fuhrmann and Wang [1988]:

$$E[R_i] \approx \frac{E[C_{b,1}]}{E[C_{b,i}]} E[R_1]. \quad (2.5)$$

Now, from (2.2), (2.3) and (2.5) we get:

$$E[W_i] = \frac{E[C_{b,1}]E[R_1]/E[C_{b,i}] + E[C_{b,i}](k_i^{(2)} - k_i)/(2k_i)}{1 - k_i \lambda_i E[C_{b,i}]}. \quad (2.6)$$

Next we recall from Boxma [1989] (Eqs. (3.21) and (3.27)) the conservation law for cyclic polling system with limited-1 service and bulk arrivals. Expressing that law in our notation yields:

$$\begin{aligned} \sum_{i=1}^N \rho_i \left[1 - \frac{\lambda_i k_i s}{1 - \rho} \right] E[W_i] &= \rho \sum_{i=1}^N \frac{\lambda_i k_i b_i^{(2)}}{2(1 - \rho)} + \sum_{i=1}^N \frac{\lambda_i b_i (k_i^{(2)} - k_i)(b_i + s)}{2(1 - \rho)} \\ &\quad + \rho \frac{s^{(2)}}{2s} + \frac{s}{2(1 - \rho)} [\rho^2 + \sum_{i=1}^N \rho_i^2]. \end{aligned} \quad (2.7)$$

From (2.6) and (2.7) one may easily solve for $E[W_i]$ by first solving for $E[R_1]$. Having derived $E[W_i]$ we now calculate the mean waiting time of the last customer in a bulk at queue i , $E[W_i^L]$. Similarly to the derivation of (2.2) we get:

$$E[W_i^L] = E[R_i] + \left[E[X_i] + k_i - 1 \right] E[C_{b,i}],$$

which yields

$$E[W_i^L] = E[W_i] + \left[k_i - 1 - \frac{k_i^{(2)} - k_i}{2k_i} \right] E[C_{b,i}]. \quad (2.8)$$

Finally, the last measure of interest is the *sojourn time* of the last customer in a bulk at queue i , $E[T_i^L]$. This value is identical to the mean sojourn time of the whole bulk, $E[T_i^B]$, which represents the sojourn time of a transaction in the Transaction Driven Computer System. We get this value simply from (2.8):

$$E[T_i^B] = E[T_i^L] = E[W_i^L] + b_i. \quad (2.9)$$

Remark 2.1

Note that when the distribution of K_i is the shifted-geometric distribution then $(k_i^{(2)} - k_i)/2k_i = k_i - 1$ which implies that $E[W_i^L] = E[W_i]$.

3. Numerical Results

In this section we evaluate the quality of the approximation method by comparing it to simulation results. The conservation-law based approximations proposed in recent years for limited-1 polling systems are considered to be quite accurate at low and medium loads and less accurate at high loads (see, Boxma and Meister [1986,1987], Fuhrmann and Wang [1988], Groenendijk [1988a,1989], Srinivasan [1988]). In addition, the quality of those approximations increases when the system becomes more symmetric (note that the use of the pseudoconservation law implies that they are *exact* when the system is fully symmetric).

Being based on a similar approach, our approximation resembles those methods in its properties; namely, it is more sensitive in cases which combine high system load ($\rho > 0.7$) with high degree of asymmetry. An examination of various cases with low and medium loads (some are reported in Levy [1988]) supports this observation.

For this reason and for the sake of brevity we focus in this comparison on examining cases which are likely to be "problematic", namely cases with high load ($\rho = 0.8, 0.9$), and avoid presenting extensive results for low and medium load cases. Therefore, the results reported in this section should be considered by the reader with the appropriate caution, keeping in mind that a more complete examination of the parameter space should yield more accurate results.

The comparisons are conducted as follows: In all cases considered we approximate the mean bulk sojourn time ($E[T_i^B]$) and compare it to simulation results. Each simulation was run with 3,000,000 customers. The accuracy of the simulation results can be verified by applying the left-hand side of the conservation law to the mean waiting times observed in the simulation ($E[W_i]$) and comparing them to the right-hand-side of the conservation law as computed from the system parameters. The relative error between these two measures is bounded in all cases by three percent.

An outline of the cases and their main characteristics is provided in Table 3.1. The cases vary from each other in the number of queues considered, the total load, the bulk size distributions and the homogeneity of individual loads, individual arrival rates, and mean bulk sizes. The table organizes the different cases according to these parameters. The last column of the table summarizes the quality of the approximation.

The details of all cases appear in the appendix. For each case we provide a detailed list of the parameters as well as the results of the approximation and of the simulation (an additional column labeled "Elimination" appears in some of the tables and refers to the elimination procedure discussed in the next section).

3.1. Discussion of the Results

When examining the results summarized in the appendix one should be careful in interpreting the accuracy of the mean sojourn times as observed at small queues. At first glance it may seem that the relative error in predicting these measures is very high. For example, the error in predicting the mean sojourn time of the small queues in Case 9b is 0.84 and the relative error is therefore $0.84/4.64 = 0.18$. However, the error of 0.84 is very small ($0.84/40.86 = 0.02$) when considered with respect to the other delay figures of the system. A more extensive discussion of this issue can be found in Whitt [1985]. The actual selection of the right accuracy measure obviously depends on the specific application.

From the results we may conclude that the main factors affecting the accuracy of the approximation are 1) The total system load, and 2) The homogeneity of the individual loads. We observe that the cases in which the approximation does not perform well are those in which the system-load is high and the individual loads significantly differ from each other (e.g., 7a-c). In all other cases (either low to medium system-load or relatively homogeneous individual loads) the approximation performs quite well.

Note that the degree of homogeneity of either the *arrival rates* or of the *bulk sizes* does not significantly affect the quality of the approximation, provided that the *individual loads* are kept relatively homogeneous. This observation is supported by Cases 5a-c and 6a-d.

In the next section we examine an elimination procedure which is designed to improve the accuracy of the approximation method in the more difficult cases, namely those which combine high system load with high degree of non-homogeneity of individual loads.

| Case | # Stations | Total Load (ρ) | Indiv. Loads (ρ_i) | Indiv. Arr. Rates | Indiv. Bulk Sizes | Comments | Quality |
|------|------------|-----------------------|---------------------------|-------------------|-------------------|--|---------|
| 1a-c | 3 | high | 1 high, 2 low | equal | non-equal | various bulk-size distributions | fair |
| 2a-b | 3 | high & very high | 1 high, 2 low | non-equal | equal | | fair |
| 3 | 3 | moderate | 1 high, 2 low | equal | non-equal | very asymmetric with respect to bulk size | good |
| 4 | 3 | high | 1 low, 1 med., 1 high | non-equal | equal | queues differ on bulk size distributions | fair |
| 5a-c | 3 | high | equal | non-equal | non-equal | various combinations of bulk-size distributions | good |
| 6a-d | 6 | high & very high | equal | non-equal | non-equal | various degrees of non-homogeneity (with respect to mean bulk sizes and arrival rates) | good |
| 7a-c | 16 | high | 4 high, 12 low | equal | non-equal | various structures (with respect to pattern of parameters). | poor |
| 8 | 16 | high | 1 high, 15 low | non-equal | equal | | good |
| 9a-b | 16 | high | 2 high, 14 low | non-equal | non-equal | combination of various mean bulk sizes and various bulk size distributions | good |

TABLE 3.1: Summary of cases

4. Elimination Procedure

As can be seen from the numerical results, the hardest cases to approximate are those which combine high system load with high degree of non-homogeneity of the individual loads $\rho_i = \lambda_i k_i$. Note, that in such cases certain queues can be distinguished as having considerably higher loads than others. The purpose of this section is to investigate an approach which eliminates one or more of such 'big' queues from the system and replaces them by a switch-over time. The reasoning behind this approach is that the reduced system is more symmetric and is - hopefully - easier to approximate. This idea is not new; it has been applied before for systems with non-zero switch-over periods, single Poisson arrivals and limited-1 service. Watson [1985] discussed the possibility of replacing a heavily-loaded queue by a switch-over time and analyzed the mean delay at the other queues using the conservation law. Boxma and Meister [1986] considered cases in which one heavily loaded queue, say queue i , dominates the system. They proposed to analyze the mean delays in this system by replacing the heavily loaded queue by a switch-over time. This switch-over time is set to zero with probability α_i and to B_i (the service time at queue i) with probability $1 - \alpha_i$; here α_i denotes the probability that at a

polling instant of queue i the server finds that queue empty. Their approach is to first approximate the original system and obtain the mean waiting time for the heavily loaded queue. Then this queue is removed from the system and replaced by a switch-over time. Now in the reduced system the mean waiting times are approximated, yielding the mean waiting times at the smaller queues. Note that the resulting approximation does not satisfy the conservation law. Boxma and Meister recommend application of their elimination procedure when 1) $\rho \geq 0.7$, 2) the total switch-over time is not negligible, and 3) the arrival rates at a small group of queues are at least three times as high as at any of the other queues. Note that that approach cannot be applied to TDCS's since in their modeling we assume zero switch-over times. As suggested in Groenendijk [1989] one can derive the mean waiting time for the eliminated queue by a back-substitution of the mean waiting times obtained from the elimination procedure into the conservation law.

In Section 4.1 we propose a new elimination procedure based on the ideas introduced by Boxma and Meister [1986]. The main problem there is to find a reasonable expression for the switch-over time replacing the eliminated queue. In Section 4.2 we describe how to apply the elimination procedure for cases consisting of *several* heavily loaded queues. In Section 4.3 we evaluate the quality of the procedure by examining numerical results. Guidelines for application of the different procedures are presented in Section 4.4.

4.1. Description of the Basic Elimination Procedure

Consider a cyclic polling model with N queues Q_1, \dots, Q_N as described in the beginning of this paper. Suppose all switch-over times are identically zero. Let τ_0 be a departure instant (of a customer) at which the system becomes empty. Let τ_1 be the first arrival instant after τ_0 . Note that the customer arriving at τ_1 finds an empty system, is immediately taken into service and initiates a number of 'busy' cycles; denote the mean number of such cycles by K_B . Let τ_2 be the instant at which the system becomes empty again. Note that, with $\Lambda := \sum_{i=1}^N \lambda_i$:

$$E[\tau_2 - \tau_1] = \frac{\rho}{\Lambda(1-\rho)}, \quad (4.1)$$

which is the mean length of a busy period in an M/G/1 queue with arrival intensity Λ and traffic intensity ρ . Note that $E[\tau_1 - \tau_0] = 1/\Lambda$. We call the time from τ_0 to τ_2 a *super-cycle* and denote its mean length by C^{sup} ; obviously,

$$C^{sup} = E[\tau_2 - \tau_0] = \frac{1}{\Lambda} + \frac{\rho}{\Lambda(1-\rho)} = \frac{1}{\Lambda(1-\rho)} \quad (4.2)$$

Suppose we want to eliminate Q_i . The proportion of time the server is visiting Q_i during a super-cycle is $\rho_i C^{sup}$. Hence the mean visit time V_i at Q_i during a 'busy' cycle is

$$V_i = \frac{\rho_i C^{sup}}{K_B}. \quad (4.3)$$

Now we would like to replace Q_i by a switch-over period with mean s_i . We propose to approximate s_i by setting it equal to V_i . In (4.3) we still need to approximate K_B , the mean number of 'busy' cycles during a super-cycle. For that purpose we consider the following inequalities.

$$\max_{j=1 \dots N} \{\lambda_j k_j\} C^{sup} \leq K_B \leq \sum_{j=1}^N \lambda_j k_j C^{sup}. \quad (4.4)$$

The explanation of these inequalities is as follows: $\lambda_j k_j C^{sup}$ is the average number of type- j customer arrivals during a super-cycle. Using the same balance argument as above this is also equal to the

average number of type- j services during a super-cycle. Since the service strategy is limited-1, we should have on the average at least as many ‘busy’ cycles during a super-cycle as $\max_{j=1\dots N}\{\lambda_j k_j\}$. This is the motivation for the inequality on the left. To motivate the inequality on the right, note that $\sum_{j=1}^N \lambda_j k_j$ represents the average total number of arrivals (and hence also of departures) during a super-cycle. Since some of these departures will be from the same ‘busy’ cycle, the average number of ‘busy’ cycles in a super-cycle should clearly be less than $\sum_{j=1}^N \lambda_j k_j$. Finally, since K_B can be anywhere between the left- and right-hand side of (4.4), we propose to approximate it by

$$K_B \approx \frac{\max_{j=1\dots N}\{\lambda_j k_j\} + \sum_{j=1}^N \lambda_j k_j}{2} \times C^{sup}; \quad (4.5)$$

we have no good feeling whether some weighted sum would perform better here. Combining (4.3) and (4.5) yields our approximation for the switch-over time s_i :

$$s_i := \frac{2\rho_i}{\max_{j=1\dots N}\rho_j + \rho}. \quad (4.6)$$

In addition, we assume that the second moment of the switch-over time is the same as the first moment, and hence $s_i^{(2)} = s_i$. The motivation for this is that in a ‘busy’ cycle the visit time at Q_i is either equal to 0 or to 1; hence if the expected visit time at Q_i in a ‘busy’ cycle is $\alpha_i = (1 - \alpha_i)0 + \alpha_i 1$, then the second moment is $(1 - \alpha_i)0^2 + \alpha_i 1^2$ and hence also equal to α_i .

4.2. Several ‘Big’ Stations: Averaging of Several Eliminations

It is natural to apply the ‘basic’ elimination procedure when there is only one heavily-loaded station dominating several smaller ones. When there are several heavily loaded stations, it is not clear from the previous section which station should be eliminated. Suppose there are $L > 1$ heavily-loaded stations, without loss of generality assume Q_1, \dots, Q_L are heavily loaded, the other queues are lightly loaded. In such a case we suggest to apply the following *averaging* procedure: first eliminate queue 1 and approximate the mean waiting times $E[W^{(1)}], \dots, E[W_N^{(1)}]$ as described above. Next do the same for Q_2 , yielding approximations $E[W^{(2)}], \dots, E[W_N^{(2)}]$. Repeat this process up to the last of the heavily-loaded queues, yielding approximations for $E[W^{(L)}], \dots, E[W_N^{(L)}]$. The final step is to average over these mean waiting times:

$$E[W_i] = \frac{1}{L} \sum_{j=1}^L E[W_i^{(j)}], \quad i = 1, 2, \dots, N.$$

In other words, the averaging procedure applies the basic elimination procedure for each of the L heavily-loaded stations; then the results are averaged.

4.3. Guidelines and Recommendations for Applying the Elimination Procedure

The following guidelines result from the experience gained in applying the elimination procedure. The first guideline is that the elimination procedure should not be applied unless the system load is above $\rho = 0.6 - 0.7$. In such cases ($\rho > 0.6 - 0.7$) application of the elimination procedure is recommended only if the arrival rates $\lambda_i k_i$, $i = 1, 2, \dots, N$ are quite asymmetric. A good rule of thumb is to apply elimination when the highest individual load is at least twice as high as the lowest individual load.¹ The second guideline concerns cases in which there are several heavily-loaded stations which

1. (See the discussion in Section 5.)

are *relatively similar*, i.e., the difference in load between these stations is smaller than, say, 0.1. In such cases we recommend to apply the averaging procedure.

4.4. Numerical Results

In this section we discuss the application of the elimination procedure to the cases given in the appendix. In parentheses we indicate the recommendations according to Section 4.3.

We start with Table A.1. In Case 1a, the quality of the two methods turns out to be about the same (Guidelines: do not apply elimination). In Cases 1b and 1c the elimination procedure is better than the basic approximation (Guidelines: apply elimination). This last observation also applies to Cases 2a and 2b in Table A.2.

In Table A.3 the total load is 0.51 and elimination was not considered. The basic approximation is of good quality here. In Table A.4 we have one heavily-loaded queue, one moderately-loaded queue and one lightly-loaded queue, with loads of 0.4, 0.266 and 0.133 respectively. The basic approximation is quite poor and the elimination yields very good results (Guidelines: apply elimination).

In Tables A.5 and A.6, all queues carry exactly the same load, and the quality of the basic approximation is good (Guidelines: do not apply elimination). We have tried the elimination procedure here (see discussion below) and it proved to be not good. In Table A.7 we have 3 cases with 4 heavily-loaded queues, each with a load of 0.128, and 12 lightly-loaded queues, each with a load of 0.024. The basic approximation does not handle these cases very well while the elimination procedure produces very good results (Guidelines: apply elimination with averaging). In all three cases we applied the averaging procedure as described in Section 4.2. We eliminated each of the stations 1-4 once and averaged over the results.

In Table A.8 station 1 was eliminated. The results of the basic approximation are good, while those of the elimination procedure are excellent (Guidelines: apply elimination). In Table A.9 we applied the averaging procedure in both Case 9a and 9b and eliminated the heavily-loaded queues 1 and 7. The observations of Table A.8 apply here as well.

In all the cases in which our guidelines suggested application of the elimination procedure, the results improved on those of the basic approximation. One may thus be tempted to try and apply the elimination procedure also in cases where the conditions for the guidelines are not met. However, this application is risky: while the results can improve in several cases they may significantly degrade in others. We shall illustrate this using Cases 5 and 6. In both cases all stations in the system carry the same load and the basic approximation produces good results. Applying the elimination procedure to these cases (single elimination for Cases 5a-c and two eliminations with averaging to Cases 6a-d) we obtain the following results:

For Table A.5:

| Variables | Case | Elimination | Case | Elimination | Case | Elimination |
|------------|------|-------------|------|-------------|------|-------------|
| $E[T_1^B]$ | 5a | 18.16 | 5b | 17.41 | 5c | 24.50 |
| $E[T_2^B]$ | | 45.51 | | 33.86 | | 47.02 |
| $E[T_3^B]$ | | 51.06 | | 50.31 | | 69.52 |

For Table A.6:

| Variables | Case | Elimination | Case | Elimination |
|----------------|------|-------------|------|-------------|
| $E[T_{1,2}^B]$ | 6a | 19.58 | 6c | 6.52 |
| $E[T_{3,4}^B]$ | | 37.00 | | 37.01 |
| $E[T_{5,6}^B]$ | | 54.42 | | 67.50 |
| $E[T_{1,2}^B]$ | 6b | 49.68 | 6d | 31.05 |
| $E[T_{3,4}^B]$ | | 74.50 | | 74.50 |
| $E[T_{5,6}^B]$ | | 99.33 | | 117.95 |

Comparing these results with the simulation results in the tables, we see that the results for Cases 5a-c are still good, but the results for Cases 6a-d are quite poor, especially for the Cases 6b and 6d, in which $\rho=0.9$. These examples show that one should be very careful when applying the elimination procedure in such non-recommended cases.

A possible way to combine the basic approximation and the elimination results is to use a linear combination of the results in which the weight of elimination results *decreases* with the degree of symmetry. Specifically, denote by $\rho_{\max} := \max_{i=1 \dots N} \rho_i$ and $\rho_{\min} := \min_{i=1 \dots N} \rho_i$ the maximum and minimum value of ρ_i respectively. Then a robust approximation for the mean waiting times would be:

$$E[W_i] := \frac{\rho_{\min}}{\rho_{\max}} E[W_i^{app}] + \frac{\rho_{\max} - \rho_{\min}}{\rho_{\max}} E[W_i^{elim}].$$

Here $E[W_i^{app}]$ denotes the mean waiting time for station i as obtained from the basic approximation and $E[W_i^{elim}]$ denotes the mean waiting time as obtained via the elimination procedure.

5. Concluding Remarks, Applications and Extensions

In this paper we suggested a model for Transaction Driven Computer Systems. We modeled a TDCS by a cyclic polling system with bulk arrivals and limited-1 service and provided an approximate analysis of the mean sojourn time in the system. The approximation proves to be quite accurate for a wide range of parameters.

A key issue raised in the design of TDCS is the proper allocation of processor time between the different programs. A common practice is to assign more than one process to perform a program which requires a large share of the processor time. The effect of such assignments on the system performance is therefore important for the proper design of the system.

The various design alternatives of process assignment can be accommodated within the framework of our model by a proper use of the service time and bulk size parameters. To demonstrate this, consider a transaction whose service time is T_i time units and assume that l_i processes are assigned to process type- i transactions. This implies that the number of time units granted to a type- i transaction in every cycle is about l_i . This transaction will be represented in the polling model by a bulk consisting of T_i/l_i customers, the service time of each of them is l_i units.

An alternative approach for modeling this system is to use a polling system with limited- l_i service. This, however, requires the analysis of the latter (a polling system with bulk arrivals and limited- l_i service) which is a topic for future research.

Another topic for future research is the analysis of systems with *transaction routing*. In these systems a transaction may be routed to be processed by program j after being processed by program i . The performance of such systems can be very sensitive to the allocation of the processor time. The analysis of these systems may be assisted by the analysis of polling systems with customer routing (Boxma [1989], Sidi and Levy [1988]).

REFERENCES

- Boxma, O.J. (1989). Workloads and waiting times in single-server systems with multiple customer classes. *To appear in Queueing Systems, Vol. 5.*
- Boxma, O.J., Groenendijk, W.P. (1987). Pseudo-conservation laws in cyclic-service systems. *J. Appl. Prob.* **25**, 949-964.
- Boxma, O.J., Meister, B. (1986). Waiting-time approximations for cyclic-service systems with switch-over times. *Performance Evaluation Review* **14**, 254-262.
- Boxma, O.J., Meister, B. (1987). Waiting-times in multi-queue systems with cyclic service. *Performance Evaluation* **7**, 59-70.
- Ferguson, M.J., Aminetzah, Y.J. (1985). Exact results for nonsymmetric token ring systems. *IEEE Trans. Commun.*, Vol. COM-33, 223-231.
- Fuhrmann, S.W., Wang, Y.T. (1988). Mean waiting time approximations of cyclic service systems with limited service. In: *Performance '87*, eds. P.-J. Courtois and G. Latouche, North-Holland Publ. Cy., Amsterdam, 253-265.
- Groenendijk, W.P. (1988). A conservation-law based approximation algorithm for waiting times in polling systems. *Report Centre for Mathematics and Computer Science, Amsterdam.*
- Groenendijk, W.P. (1989). Waiting-time approximations for cyclic-service systems with mixed service strategies. In: *Proc. 12th ITC*, North-Holland Publ. Cy., Amsterdam.
- Kühn, P.J. (1979). Multiqueue systems with nonexhaustive cyclic service. *The Bell System Technical Journal*, Vol. **58**, 671-698.
- Leung, K.K. (1988). Performance analysis of a processor sharing policy with interactive and background jobs. *Report AT&T Bell Labs, Holmdel.*
- Levy, H. (1988). Single-processor transaction oriented systems: modeling and analysis. *Report AT&T Bell Labs, Holmdel.*
- Sidi, M., Levy, H. (1988). A queueing network with a single cyclically roving server. *Report Department of Computer Science, Tel-Aviv University.*
- Srinivasan, M.M. (1988). An approximation for mean waiting times in cyclic server systems with nonexhaustive service. *Performance Evaluation* **9**, 17-33.
- Watson, K. S. (1985). Performance evaluation of cyclic service strategies - a survey. In: *Performance '84*, ed. E. Gelenbe, North-Holland Publ. Cy., Amsterdam, 521-533.
- Whitt, W. (1985). Approximations for the GI/G/m queue. *Report AT&T Bell Labs, Murray Hill.*
- Wolff, R.W. (1982). Poisson arrivals see time averages. *Oper. Res.* **30**, 223-231.

APPENDIX: Detailed results (mean bulk sojourn time)

TABLE A.1

3 queues, high load ($\rho=0.8$), symmetric arrivals, asymmetric bulk sizes.

Arrival Rates: $\lambda_1=\lambda_2=\lambda_3=0.03333$.

Bulk Sizes:

Case 1a: $k_1=10, k_2=k_3=7, k_1^{(2)}=190, k_2^{(2)}=k_3^{(2)}=91$

Case 1b: $k_1=14, k_2=k_3=5, k_1^{(2)}=378, k_2^{(2)}=45$

Case 1c: $k_1=14, k_2=k_3=5, k_1^{(2)}=196, k_2^{(2)}=k_3^{(2)}=25$

| Case | Parameters | Variables | Simulation | Approximation | Elimination |
|------|---|----------------|------------|---------------|-------------|
| 1a | $k_1=10, k_2=k_3=7,$ shifted geometric | $E[T_1^B]$ | 52.88 | 48.48 | 56.59 |
| | | $E[T_{2,3}^B]$ | 28.76 | 32.63 | 26.84 |
| 1b | $k_1=14, k_2=k_3=5,$ shifted geometric | $E[T_1^B]$ | 74.26 | 71.10 | 73.66 |
| | | $E[T_{2,3}^B]$ | 14.93 | 18.60 | 15.03 |
| 1c | $k_1=14, k_2=k_3=5,$ deterministic | $E[T_1^B]$ | 49.28 | 47.20 | 48.34 |
| | | $E[T_{2,3}^B]$ | 12.51 | 15.04 | 12.81 |

TABLE A.2

3 queues, high and very high loads ($\rho=0.8-0.9$), asymmetric arrivals, symmetric bulk sizes.

Arrival Rates: $\lambda_1=0.06, \lambda_2=\lambda_3=0.02$.

Bulk Sizes:

Case 2a: $k_1=k_2=k_3=8, k_1^{(2)}=k_2^{(2)}=k_3^{(2)}=120$ (shifted geometric)

Case 2b: $k_1=k_2=k_3=9, k_1^{(2)}=k_2^{(2)}=k_3^{(2)}=153$ (shifted geometric)

| Case | Parameters | Variables | Simulation | Approximation | Elimination |
|------|---|----------------|------------|---------------|-------------|
| 2a | high load: $\rho=0.8$ $k_1=k_2=k_3=8$ | $E[T_1^B]$ | 47.22 | 43.29 | 47.27 |
| | | $E[T_{2,3}^B]$ | 24.13 | 30.06 | 24.09 |
| 2b | very high load: $\rho=0.9$ $k_1=k_2=k_3=9$ | $E[T_1^B]$ | 118.25 | 114.64 | 122.42 |
| | | $E[T_{2,3}^B]$ | 33.91 | 41.79 | 30.11 |

TABLE A.3

3 queues, moderate load ($\rho=0.51$), symmetric arrivals, very asymmetric bulk sizes.

Arrival Rates: $\lambda_1=\lambda_2=\lambda_3=0.009091$

Bulk Sizes: $k_1=40, k_2=k_3=4, k_1^{(2)}=3160, k_2^{(2)}=k_3^{(2)}=28$

| Case | Parameters | Variables | Simulation | Approximation |
|------|---|----------------|------------|---------------|
| 3 | $k_1=40, k_2=k_3=8,$ shifted geometric | $E[T_1^B]$ | 83.87 | 81.06 |
| | | $E[T_{2,3}^B]$ | 14.09 | 15.53 |

TABLE A.4

3 queues, high load ($\rho=0.8$), symmetric arrivals, asymmetric bulk sizes.

Arrival Rates: $\lambda_1=\lambda_2=\lambda_3=0.02667$

Bulk Sizes: $k_1=5, k_2=10, k_3=15, k_1^{(2)}=25, k_2^{(2)}=190, k_3^{(2)}=225$

| Case | Parameters | Variables | Simulation | Approximation | Elimination |
|------|----------------------------|------------|------------|---------------|-------------|
| 4 | $k_1=5$ deterministic | $E[T_1^B]$ | 12.77 | 15.84 | 13.16 |
| | $k_2=10$ shifted geometric | $E[T_2^B]$ | 40.93 | 47.82 | 38.38 |
| | $k_3=15$ deterministic | $E[T_3^B]$ | 56.38 | 47.65 | 57.58 |

TABLE A.5

3 queues, high load ($\rho=0.8$), asymmetric arrivals, asymmetric bulk sizes.

Arrival Rates: $\lambda_1=0.05333, \lambda_2=0.02667, \lambda_3=0.01778$

Bulk Sizes:

Case 5a: $k_1=5, k_2=10, k_3=15, k_1^{(2)}=25, k_2^{(2)}=190, k_3^{(2)}=225$

Case 5b: $k_1=5, k_2=10, k_3=15, k_1^{(2)}=25, k_2^{(2)}=100, k_3^{(2)}=225$

Case 5c: $k_1=5, k_2=10, k_3=15, k_1^{(2)}=45, k_2^{(2)}=190, k_3^{(2)}=435$

| Case | Parameters | Variables | Simulation | Approximation |
|------|----------------------------|------------|------------|---------------|
| 5a | $k_1=5$ deterministic | $E[T_1^B]$ | 18.00 | 17.29 |
| | $k_2=10$ shifted geometric | $E[T_2^B]$ | 46.26 | 48.01 |
| | $k_3=15$ deterministic | $E[T_3^B]$ | 50.86 | 53.01 |
| 5b | $k_1=5$ deterministic | $E[T_1^B]$ | 17.90 | 17.29 |
| | $k_2=10$ deterministic | $E[T_2^B]$ | 34.21 | 35.15 |
| | $k_3=15$ deterministic | $E[T_3^B]$ | 50.97 | 53.01 |
| 5c | $k_1=5$ shifted geometric | $E[T_1^B]$ | 25.04 | 23.01 |
| | $k_2=10$ shifted geometric | $E[T_2^B]$ | 49.52 | 48.01 |
| | $k_3=15$ shifted geometric | $E[T_3^B]$ | 73.19 | 73.02 |

TABLE A.6

6 queues, high and very high loads ($\rho=0.8-0.9$), asymmetric arrivals, asymmetric bulk sizes.
All bulk sizes shifted geometric.

Case 6a:

$$\lambda_1=\lambda_2=0.03333, \lambda_3=\lambda_4=0.01667, \lambda_5=\lambda_6=0.01111$$

$$k_1=k_2=4, k_3=k_4=8, k_5=k_6=12, k_1^{(2)}=k_2^{(2)}=28, k_3^{(2)}=k_4^{(2)}=120, k_5^{(2)}=k_6^{(2)}=276$$

Case 6b:

$$\lambda_1=\lambda_2=0.0375, \lambda_3=\lambda_4=0.01875, \lambda_5=\lambda_6=0.0125$$

$$k_1=k_2=4, k_3=k_4=8, k_5=k_6=12, k_1^{(2)}=k_2^{(2)}=28, k_3^{(2)}=k_4^{(2)}=120, k_5^{(2)}=k_6^{(2)}=276$$

Case 6c:

$$\lambda_1=\lambda_2=0.1333, \lambda_3=\lambda_4=0.01667, \lambda_5=\lambda_6=0.008889$$

$$k_1=k_2=1, k_3=k_4=8, k_5=k_6=15, k_1^{(2)}=k_2^{(2)}=1, k_3^{(2)}=k_4^{(2)}=120, k_5^{(2)}=k_6^{(2)}=435$$

Case 6d:

$$\lambda_1=\lambda_2=0.15, \lambda_3=\lambda_4=0.01875, \lambda_5=\lambda_6=0.01$$

$$k_1=k_2=1, k_3=k_4=8, k_5=k_6=15, k_1^{(2)}=k_2^{(2)}=1, k_3^{(2)}=k_4^{(2)}=120, k_5^{(2)}=k_6^{(2)}=435$$

| Case | Parameters | Variables | Simulation | Approximation |
|------|----------------------------|----------------|------------|---------------|
| 6a | high load: $\rho=0.8$ | | | |
| | $k_1=k_2=4$ | $E[T_{1,2}^B]$ | 18.54 | 18.00 |
| | $k_3=k_4=8$ | $E[T_{3,4}^B]$ | 37.87 | 38.00 |
| | $k_5=k_6=12$ | $E[T_{5,6}^B]$ | 56.51 | 58.00 |
| 6b | very high load: $\rho=0.9$ | | | |
| | $k_1=k_2=4$ | $E[T_{1,2}^B]$ | 36.91 | 35.50 |
| | $k_3=k_4=8$ | $E[T_{3,4}^B]$ | 74.71 | 75.50 |
| | $k_5=k_6=12$ | $E[T_{5,6}^B]$ | 110.19 | 115.50 |
| 6c | high load: $\rho=0.8$ | | | |
| | $k_1=k_2=1$ | $E[T_{1,2}^B]$ | 3.47 | 3.00 |
| | $k_3=k_4=8$ | $E[T_{3,4}^B]$ | 38.15 | 38.00 |
| | $k_5=k_6=15$ | $E[T_{5,6}^B]$ | 72.58 | 73.02 |
| 6d | very high load: $\rho=0.9$ | | | |
| | $k_1=k_2=1$ | $E[T_{1,2}^B]$ | 6.91 | 5.50 |
| | $k_3=k_4=8$ | $E[T_{3,4}^B]$ | 75.43 | 75.50 |
| | $k_5=k_6=15$ | $E[T_{5,6}^B]$ | 146.65 | 145.50 |

TABLE A.7

16 queues, high load ($\rho=0.8$), asymmetric arrivals, symmetric bulk sizes.

Arrival Rates: $\lambda_1 = \dots = \lambda_4 = 0.016$, $\lambda_5 = \dots = \lambda_{16} = 0.003$

Bulk Sizes:

Case 7a: $k_1 = \dots = k_{16} = 8$, $k_1^{(2)} = \dots = k_{16}^{(2)} = 120$

Case 7b: $k_1 = \dots = k_{16} = 8$, $k_1^{(2)} = \dots = k_{16}^{(2)} = 64$

Case 7c: $k_1 = \dots = k_{16} = 8$, $k_i^{(2)} = 64$ for i odd, $k_i^{(2)} = 120$ for i even

| Case | Parameters | Variables | Simulation | Approximation | Elimination |
|------|--|----------------------------|------------|---------------|-------------|
| 7a | $k_1 = k_2 = \dots = k_{16} = 8$, shifted geometric | $E[T_{1-4}^B]$ | 43.02 | 38.48 | 42.42 |
| | | $E[T_{5-16}^B]$ | 29.11 | 37.15 | 30.14 |
| 7b | $k_1 = k_2 = \dots = k_{16} = 8$, deterministic | $E[T_{1-4}^B]$ | 33.75 | 31.65 | 32.90 |
| | | $E[T_{5-16}^B]$ | 27.50 | 35.28 | 28.94 |
| 7c | $k_1 = k_2 = \dots = k_{16} = 8$, i odd: shifted geometric, i even: deterministic | $E[T_{1,3}^B]$ | 34.96 | 31.65 | 35.04 |
| | | $E[T_{2,4}^B]$ | 42.16 | 38.48 | 40.29 |
| | | $E[T_{5,7,9,11,13,15}^B]$ | 27.68 | 35.28 | 28.94 |
| | | $E[T_{6,8,10,12,14,16}^B]$ | 29.39 | 37.15 | 30.14 |

TABLE A.8

16 queues, high load ($\rho=0.8$), asymmetric arrivals, symmetric bulk sizes.

Arrival Rates: $\lambda_1 = 0.06$, $\lambda_2 = \dots = \lambda_{16} = 0.002667$

Bulk Sizes: $k_1 = \dots = k_{16} = 8$, $k_1^{(2)} = \dots = k_{16}^{(2)} = 120$

| Case | Parameters | Variables | Simulation | Approximation | Elimination |
|------|---|-----------------|------------|---------------|-------------|
| 8 | $\lambda_1 = 0.06, \lambda_2 = \dots = \lambda_{16} = 0.002667$, $k_1 = \dots = k_{16} = 8$ | $E[T_1^B]$ | 49.78 | 47.29 | 49.75 |
| | | $E[T_{2-16}^B]$ | 20.27 | 24.08 | 20.40 |

TABLE A.9

16 queues, high load ($\rho=0.81$), asymmetric arrivals, asymmetric bulk sizes.

Arrival Rates: $\lambda_1=\lambda_7=0.05$, $\lambda_2=\dots=\lambda_6=\lambda_8=\dots=\lambda_{16}=0.015$

Bulk Sizes: (Station 1 - shifted geometric, station 7 - two-peak distribution ($\Pr[1] = 3/4$, $\Pr[21] = 1/4$), other stations - deterministic)

Case 9a: $k_1=k_7=6$, $k_2=\dots=k_6=k_8=\dots=k_{16}=1$,
 $k_1^{(2)}=66$, $k_7^{(2)}=111$, $k_2^{(2)}=\dots=k_6^{(2)}=k_8^{(2)}=\dots=k_{16}^{(2)}$

Case 9b: $k_1=k_7=6$, $k_2=\dots=k_6=k_8=\dots=k_{16}=2$,
 $k_1^{(2)}=66$, $k_7^{(2)}=111$, $k_2^{(2)}=\dots=k_6^{(2)}=k_8^{(2)}=\dots=k_{16}^{(2)}$

| Case | Parameters | Variables | Simulation | Approximation | Elimination |
|------|---|---------------------|------------|---------------|-------------|
| 9a | $k_1=k_7=6$, $k_2=\dots=k_6=k_8=\dots=k_{16}=1$ | $E[T_1^B]$ | 30.84 | 29.98 | 30.98 |
| | | $E[T_7^B]$ | 41.30 | 42.07 | 41.12 |
| | | $E[T_{2-6,8-16}^B]$ | 1.91 | 1.60 | 1.77 |
| 9b | $k_1=k_7=6$, $k_2=\dots=k_6=k_8=\dots=k_{16}=2$ | $E[T_1^B]$ | 30.20 | 30.20 | 31.34 |
| | | $E[T_7^B]$ | 40.86 | 42.28 | 41.48 |
| | | $E[T_{2-6,8-16}^B]$ | 4.64 | 5.48 | 4.89 |