CONF-771107--1

LA-UR -77-893

TITLE: The Effects of Resource Request Patterns in Queueing Network Models

AUTHOR(S): Tom W. Keller

MASTER

SUBMITTED TO: 6th Annual Symposium on Operating Systems Principles, ~~Purdue University~~ West Lafayette, IN.

The Los Alamos Scientific Laboratory requests that the publisher identify this article as work performed under the auspices of the USERDA.

los alamos scientific laboratory
of the University of California
LOS ALAMOS, NEW MEXICO 87544

Form No. 836
St. No. 2629
1/75

UNITED STATES
ENERGY RESEARCH AND
DEVELOPMENT ADMINISTRATION
CONTRACT W-7405-ENG. 36

The Effects of Resource Request Patterns in Queueing Network Models

Tom W. Keller

Los Alamos Scientific Laboratory,
Los Alamos, New Mexico 87545

ABSTRACT

Queueing network models of computing systems are becoming increasingly popular [M1] because of their robustness and ease of solution. The impact of resource request patterning upon these models is presently not well understood. In this paper we prove the counter-intuitive fact that the effect of both holding time and routing patterning upon the performance measures for the important class of local balance [B1,C1,R1] models is nil. A study of a larger class of models, over a parameter space typical of models of computing systems, reveals that the impact of patterning of resource requests upon performance measures is negligible. This conclusion has important consequences for understanding the robustness of queueing network models of computing systems and for the level of detail necessary for accurate workload characterization. The practical consequences of this result to performance analysis of computing systems are considered.

1. Introduction

This paper investigates the impact of patterning of resource requests upon queueing network models of computing systems. The central question of this investigation is "How do customer resource request behavior patterns impact existing queueing network models of computing system performance?" The motivation for this research comes from the observation that jobs exhibit patterns in their behavior. One behavior "pattern" is locality in memory references [D1]. Another example is CPU burst time "modes" investigated by Lo [L1]. The increasingly popular queueing network models of computing systems for which product-form solutions hold [B1,C1,R1], are seemingly insensitive to many types of patterned job behavior, remaining accurate performance predictors of systems in which behavior patterns surely exist ([G1] and [R2], for example). Why? The answer to this question affects the robustness of existing queueing network models (and hence our confidence in their applicability and accuracy). If the impact of behavior patterns is important, then clearly the analyst must carefully characterize the workload behavior patterns, a formidable task since the practical definition of "pattern" is so broad, as the two examples above indicate. If, however, the impact of patterning is negligible, then the analyst may ignore behavior patterns and be satisfied that grosser, probabilistic, workload characterizations will suffice.

In this paper we define a means of formally modeling patterns of resource requests by customers in queueing network models, which we designate customer task graphs (ctg's). This formalism can represent a very large class of behavior patterns. We prove that for models obeying local balance (models for which product-form solutions hold), customer behavior patterns cannot impact performance measures of a model, so long as the ctg's representing arbitrary behavior patterns result in the same workloads being placed upon the servers in the model. We next consider models which are not in local balance. Since closed form solutions to these models usually do not exist, we cannot make as definitive a statement as for local balance models. A series of experiments are conducted over what are considered reasonable "worst case" computer job behavior patterns. The results of these experiments show that job behavior patterns have little impact upon the important performance metrics of the systems. The implications of these remarkable results to the systems analyst are then considered.

2. Customer Task Graphs

We define the sequencing of customer service requests by a directed graph which we label a customer task graph (ctg). The ctg is composed of n nodes, N_i, i=1,...,n. Associated with each N_i is the server S_i requested by the customer with type T_i. Let the set of servers be S, of size s, and the set of types T, of size t, such that $S_i \in S$ and $T_i \in T$, i=1,...,n. Let an arc from N_i to N_j represent the possibility of the customer next specifying a request for server S_j with type T_j. Let $q_{i,j}$ represent the non-zero probability associated with this event, such that $\sum_j q_{ij} = 1$.

Now consider a queueing network populated by M customers whose service requests are determined by a ctg, with all customers obeying the same ctg. The service time distribution at each server S_j, j=1,...,s, may vary by customer type. Chandy [C1] and Baskett, Chandy, Muntz, and Palacios [B1] have determined closed form solutions to a class of these queueing networks which have the property of local balance [C1]. Reiser and Kobayashi [R1] extended the results of [B1] to the case in which customer transitions are characterized by more than one closed Markov-subchain, and demonstrated efficient algorithms for the solution of the network. In [B1], the ctg is implicitly defined by the branching probabilities p(i,r:j,s), which is the probability of a customer of type r transiting to queue j with type s upon completion of service at queue i. We define a graph notation instead of utilizing the p(i,r:j,s) convention of [B1] because the ctg notation utilizes fewer customer types, and because the graph convention appears more natural for the representation of deterministic customer behavior patterns.

Consider the example ctg in Figure 1. The p(i,r:j,s) convention requires n_1+n_2-1 types to insure the sequencing of n_1 accesses to queue 1 followed by n_2 accesses to queue 2. The equivalent ctg requires but two types (at the expense of n_1+n_2 nodes). This example makes clear the equivalence of the two notations, as we need only to specify one type in the p(i,r:j,s) notation for each node in the ctg.

3. Proof of Equivalence of Customer Task Graphs

In this section we consider ctg queueing networks which meet the local balance condition. We prove that for every ctg queueing network with many customer types there exists an equivalent queueing network with a single type. Two queueing networks are equivalent if they have the same queue-length distributions by server, with queue length being defined as the total number of customers occupying the queue and its server. If two ctg queueing networks are equivalent to the same single-type queueing network, then they are obviously equivalent. This result falls naturally from the work in [B1], and was demonstrated in another form by Reiser and Kobayashi, in their development of efficient numerical algorithms for solution of this class of network. The consequences of this important property are discussed at the end of this section.

Consider the class of ctg queueing networks described in [B1]. In the preceding section we established that the ctg notation and the notation in [B1] are equivalent. We slightly modify the notation of [B1] for ease of exposition. We define a closed queueing network, QN, as composed of L queues, their associated servers, and M customers. There is an arbitrary but finite number of customer types R. Customers traverse the network and change type by branching probabilities p(i,r:j,s), with the transition being from queue i to queue j and from type r to type s. The service discipline at each queue is either processor-shared (PS) or first-come-first-served (FCFS). Service times for FCFS servers are assumed exponentially distributed and identical for all types. Service distributions for PS servers must have rational Laplace transforms and may vary by customer type.

Let μ_{ir}^{-1} be the mean service time for a customer of type r at server i with $\mu = \{\mu_{ir},\ i=1,\ldots,L \text{ and } r=1,\ldots R\}$. Let e_{ir} be defined by the set of linear equations

$$\sum_{L,R} e_{ir}\cdot p(i,r:j,s) = e_{js}.$$

The e_{ir} may be interpreted as the relative frequency of visits to queue i for customers of type r and are defined to within an arbitrary multiplicative constant. Closed form solutions for aggregate states of the system are derived in [B1], with the probability of the system in state S being

$$P[S=(y_1,y_2,\ldots,y_L)] = G g_1(y_1) g_2(y_2)\ldots g_L(y_L) \tag{1}$$

where $y_i=(n_{i1},n_{i2},\ldots,n_{iR})$ and n_{ir} is the number of customers in queue i of type r. G is a normalizing constant and is defined as $\sum P[S=y_1,y_2,\ldots,y_L)]$, summed over $y_1+\ldots+y_L=M$. The g_i are defined as

$$g_i(y_i) = n_i!\prod_{r=1}^{R}(1/n_{ir}!)\beta_{ir}^{n_{ir}} \tag{2}$$

where

$$\beta_{ir} = \begin{cases} c_{ir}/\mu_{ir} \text{ for PS queues} \\ c_{ir}/\mu_i \text{ for FCFS queues} \end{cases} \quad (3)$$

Let us define another marginal distribution by introducing the aggregate state $A = (n_1, n_2, \ldots, n_L)$ so that

$$P[A=(n_1,n_2,\ldots,n_L)] = Gf_1(n_1)f_2(n_2)\ldots f_L(n_L) \quad (4)$$

where n_i is the total number of customers in queue i and

$$f_i(n_i) = \sum_T g_i(y_i) \quad (5)$$

where $T = \{n_{i1}+n_{i2}+\ldots+n_{iR} = n_i\}$.

Theorem. For every queueing network QN as defined there exists an equivalent queueing network QN' with R'=1 such that for any aggregate states A and A', P[A] = P[A'].

Proof. Substituting (2) into (5) we obtain

$$f_i(n_i) = \sum_T n_i! \prod_{r=1}^{R} (1/n_{ir}!)\beta_{ir}^{n_{ir}}. \quad (6)$$

We note that (6) is a multinomial expansion, thus

$$f_i(n_i) = (\beta_{i1}+ {}_{i2}+\ldots+\beta_{iR})^{n_i} \quad (7)$$

We can define a single type for QN' such that

$$\beta_i' = \sum_{r=1}^{R} \beta_{ir}.$$

Thus $f_i'(n_i)=f_i(n_i)$ by (7). From (4) P[A] = P[A'].

The implications of this property of ctg queueing network models in local balance is profound: specific patterns in customer behavior have no impact upon the queue-length distributions of the network. The practical implication is that specific job behavior patterns play no role in determining the performance of computer systems which are adequately described by local-balance queueing network models. The importance of this property to the computer system analyst is more fully discussed in the last section of this paper.

3.1. Example.

Consider the queueing network of Figure 2. The p(i,r:j,s), e_{ir}, M, and L are defined in the preceding section. The queueing discipline is PS for both queues. β_1 and β_2 are defined in the proof. We observe that the marginal aggregate state distribution for both the original and equivalent networks are identical.

4. The Effect of Service Time Patterns in CTG Queueing Models

In the preceding section we established that specific behavior patterns did not impact queueing network models in local balance. In this section we consider the effect of specific behavior patterns upon more general, non-local balance models. The property violating local balance in the models we consider is that of allowing different service time distributions by customer type in FCFS queues. The practical motivation for considering this particular structure is the speculation that a job's service request distribution may vary according to its past history. Lo [L1] observed that CPU burst time distributions in a multiprogrammed operating system could be characterized as alternating beteen long-burst and short-burst modes. A possible interpretation of this behavior is that job CPU service requests alternate between long and short modes. We can generalize this property to include both CPU and I/O requests. CPU service disciplines are commonly modeled as PS, while I/O service disciplines are commonly modeled as FCFS. In keeping with the previous section, we evaluate the impact of ctg's with specific sequences of requests to "equivalent" (if the queueing network was in local balance) ctg's which allow arbitrary sequences of requests.

In order to keep the ctg models tractable we must restrict our evaluation to small numbers of queues and job types.

4.1. The Models

The two ctg's evaluated are displayed in Figure 3. In ctg I the customer changes type and resource cyclically. This is the "patterned" ctg. In ctg II the customer's type is chosen probabilistically while his resource requests are still cyclic. We note the queueing networks implied by these ctg's are identical. Referring to the notation of the previous section, let $\beta_{I,i}$ be defined as the sum of $\beta_{i,r}$ over all types r for ctg I. Similarly, let $e_{I,i}$ be defined for ctg I. We define $\beta_{II,i}$ and $e_{II,i}$ in the same manner. Since the e are defined to within an arbitrary multiplicative constant, for ease of exposition allow $e_{II,i}$ to be redefined as 1 for all i and r, with no loss of generality. Thus $\beta_{I,i} = \beta_{II,i}$, i=1,2. If the two ctg queueing network models were in local balance they would be equivalent, and hence their performance characteristics would be identical.

The two ctg models are evaluated for different service disciplines

(FCFS and PS), for a number of values for μ_{ir} (i=1,2 and r=1,2), and for different numbers of customers. Service times at each server are assumed exponential. For all three models the service discipline of server 2 is FCFS. For model 1 the service discipline of server 1 is FCFS while for models 2 and 3 it is PS.

4.2. Solution Method and Validation

The models were solved by an exact matrix method. The number of states for models 1, 2, and 3 are 12, 32, and 26, respectively. A property of model 2, ctg I is that the system is non-ergodic, the state space partitioning itself into two non-intersecting ergodic chains. The steady-state probabilities for states of the larger of the two chains were used in the comparison. The steady-state probabilities for states of the smaller of the two chains were not significantly different. The solution method was validated for all models by bringing them into local balance by setting $\mu_{i1}=\mu_{i2}$ for all FCFS servers i. Results were then compared against those obtained from ASQ [K1], a program for the solution of local balance models.

4.3. Results

A range of values for the mean service rates μ were chosen for the investigation of equivalence. With no loss of generality μ_{11} was set to 1. Tables 2-4 display the results of the investigation. An inspection of the tables reveals that the queue-length distributions of the two models usually agree to within 1%, with the greatest disagreement being 13%. Note that server utilizations (1-P(2) for server 1 and 1-P(0) for server 2) agree in all cases within 3.8%.

We note that the queue-length distributions for ctg's I and II of model 1 appear identical. This identity was tested by coding the solution method used for additional precision for model 1. The queue-length distributions obtained for ctg's I and II were identical to within the inherent accuracy of the solution method (at least 10^{-20}).

The close agreement of the queue-length distributions for ctg I and ctg II over the range of parameters leads us to the conclusion that the two ctg models are nearly equivalent.

5. The Effects of Branching Patterns in CTG Models

In keeping with the preceeding section we consider the impact of specific ctg patterns upon non-local balance models. Non-exponential service time distributions in FCFS queues is the property violating local balance in these models. We investigate the effect of deterministic routing through the ctg queueing networks upon the models' equivalence.

The practical motivation for this ctg structure is the observation for some general purpose computer systems that disk requests may be clustered in time. One example is a large system in which data bases are tape resident. When a job requiring a data base which is not disk resident is to be run, the appropriate data base must be copied from tape to a single disk -- resulting in a preponderant number of accesses to the disk clustered in time. When the data base must be copied back to tape the same behavior is observed. Clustering of device requests is also likely to be observed in data base systems in which data is dumped from disk to tape as part of a rollback/recovery scheme. Deterministic routing is used in the following models to enforce extreme, worst-case examples of device request clustering.

5.1. The Models

The two ctg's are displayed in Figure 4. In ctg I the customer cycles through server 0 and server 1 N_i times, then cycles through server 0 and server 2 N_i times, etc. In ctg II the customer chooses server j with probability $N_j/(N_1+N_2+N_3)$, j=1,2,3, upon completion of service at server 0. Using the notation defined in the previous section, it is intuitively obvious that $e_{I,i}=e_{II,i}$. Since it is the deterministic routing that violates local balance, let us simplify the parameter space to be explored by letting the μ be constant over customer type. As in the previous case, the two ctg models would be equivalent if local balance held. The two ctg models are evaluated for different values of $N_i(i=1,\ldots,3)$, $\mu_j(j=1,\ldots,3)$, M (the number of customers), and C_v, the coefficient of variation of the service time distribution for servers 1, 2, and 3. Two values for C_v were chosen for evaluation, 0 and $1/\sqrt{3}$. $C_v=0$ defines a constant service time and $C_v=1/\sqrt{3}$ lies between 0 and 1, with $C_v=1$ being a property of the exponential distribution. We note that for $C_v=1$ the queueing network described is in local balance if its service times are exponential and the two ctg models are known to be equivalent. With no loss of generality μ_0 is set to 1. Servers 1-3 are FCFS while server 0 may be either PS or FCFS.

A central server form [B2] for the models was chosen so as to more easily relate the possible behavior patterns for actual systems to the investigation. Thus server 0 represents the CPU, with servers 1-3 representing I/O devices. In the following sections we shall parameterize the model according to the assumptions drawn from experience with existing computer systems.

5.2. Solution Methods, Validation, and an Equivalence Metric

The state space defined by the models was too large for matrix solution methods so a simulation, coded in ASPOL [C2], was used. A Students T test approach [G2] for confidence intervals was used, as the size of the state space made the preferable Crane-Iglehart methods [C3,C4] impractical. Each run iteration of the simulation model was long enough to insure that all known distribution means (such as device holding times) were within 1% of the simulation input parameters. Values obtained from the simulation are estimated to be within ±1% of the actual values with 90% confidence. For $C_v=1/\sqrt{3}$ the actual holding time distribution was 3-stage Erlangian.

The service discipline for server 0 in the simulation is round-robin with fixed quantum (RRFQ). In the RRFQ discipline the server processes each service request for the duration of a fixed quantum. If the job's service request is completed during this quantum, it leaves the server, otherwise the job is cycled back to the end of the queue to await its next quantum of sevice. New arrivals are placed at the end of the queue. The PS discipline is approximated, when appropriate for the models, by setting the quantum to be less (by a factor of 4) than the mean service request time (μ_0^{-1}) for the server. The FCFS discipline is approximated by setting the quantum larger (by a factor of 4) than μ_0^{-1}. Realism was the motivation for using the round-robin discipline for server 0. Actual CPU service disciplines for computer systems are better approximated by round-robin than PS or FCFS, as true PS is impossible to achieve and true FCFS is usually not desirable in multiprogrammed operating systems (from the results of Sherman, Baskett, and Browne [S1]). The service disciplines for servers 1-3 are FCFS, in accordance with their role as I/O devices in the central server model.

The simulation was validated at $C_v=1$ against solutions obtained via ASQ. Server utilizations were in agreement within ±1% for all cases considered.

In contrast to the preceding sections, the equivalence metric used was not queue-length. The large amount of data which would have resulted from this approach would have been awkward to handle and would have greatly increased the expense of the simulations. Instead, an "aggregate" equivalence metric of server 0 utilization was chosen. This metric is reasonable since it is also the thruput of server 0 (since μ_0 is set at 1), and represents the work capacity of the system.

5.3. Model Parameters and Results

Table 5 displays the results of the comparison for different parameterizations of the models. For models 1-7 the discipline for server 0 is PS, for the others it is FCFS. ρ is defined by

$$\rho=(\beta_1+\beta_2+\beta_3)^{-1} \qquad (8)$$

representing the balance between server 0 and the servers 1-3 subsystem. Values for μ_1, μ_2, and μ_3 were chosen to satisfy the relationship $\mu_2^{-1}=\mu_3^{-1}=3\mu_1^{-1}$ and to satisfy the definition of ρ. The factor 3 was selected as the balance between meanservicetimes since that is typically the disparity in mean access times between disks and drums for accesses to single records. The values for N_1, N_2, and N_3 were chosen such that the condition $N_1=N_2=N_3$ reflects equal average frequencies of visits to servers 1-3 (corresponding to $e_1=e_2=e_3$) while $N_1=3N_2=3N_3$ reflects an equal workload balance between servers 1-3 ($\beta_1=\beta_2=\beta_3$). An integer value for each N_i in the table represents a ctg I model, while a dash represents a ctg II model.

Inspection of the results reveals that the two ctg models are nearly equivalent, with disparities between the equivalence metric typically less than 2% and occasionally as great as 3%. It must be remembered that the values displayed in Table 2 are approximate, and subject to simulation error within the described bounds.

6. Conclusion

The conclusion we draw is that specific resource request patterns by customers have little or no impact upon the accuracy of common queueing network models of computing systems. For the local-balance class of models we proved that patterns play no role in determining the performance measures of the models. Experiments on a wider class of models revealed that patterns had a negligible impact upon important performance measures.

The practical significance of this conclusion is that the computer system analyst can ignore resource request patterns over different workloads provided that: (1) the mean work ascribed to each resource remains constant, and (2) scheduling of service requests is ignored. Obviously, a scheduling scheme which could predict job resource requests by recognizing patterns in job behavior might result in significantly different system performances for different workloads. When the analyst is faced with a queueing network model which does not validate, these results suggest that he concentrate on causes other than job behavior patterns to account for the error.

It is tempting to draw the above conclusion for queueing network models in general. However, our investigation was limited to local balance models and models which are "close" to meeting the local balance assumptions. Thus we dare not make such a sweeping generalization. This does not detract from the usefulness of this work. The models considered were deliberately chosen because of their common usage, and hence practical value. Our conclusion is thus of great practical value to analysts working in the field.

It would be desirable to extend the investigation to wider classes of queueing network models. Further research investigating the impact of service request patterns over even larger classes of models quickly diverges according to the nature of patterns and the scheduling schemes considered.

Acknowledgements

This work was supported in part by National Science Foundation Grants GJ-1084 and DCR74-13302 while the author was at the University of Texas at Austin. Professor K. M. Chandy was responsible for many of the theories which are applied in this work. Professor J. C. Browne provided many ideas for the construction of the investigation and furnished useful insights into the application of the results.

Bibliography

[B1] 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, Vol. 22, No. 2, p. 248, April 1975.

[B2] J. P. Buzen, "Queueing Network Models of Multiprogramming," Ph.D. Thesis, Harvard University, Cambridge, Mass., 1971.

[C1] K. M. Chandy, "The Analysis and Solutions for General Queueing Networks," Proc. 6th Annual Princeton Conf. Information Sciences and Systems, Princeton Univ., Princeton, N. J., March 1972.

[C2] Control Data Corporation, A Simulation Process-Oriented Language (ASPOL) Reference Manual, C.D.C. Special Support Division, Sunnyvale, Calif., 1972.

[C3] M. A. Crane and D. I. Iglehart, "Simulating Stable Stochastic Systems, I: General Multiserver Queues," J. ACM, Vol. 21, p. 103, 1974.

[C4] M. A. Crane and D. I. Iglehart, "Simulating Stable Stochastic Systems, II: Markov Chains," J. ACM, Vol. 21, p. 114, 1974.

[D1] P. J. Denning, "Virtual Memory," Computing Surveys, Vol. 2, No. 3, pp. 153-189, September 1970.

[G1] T. P. Giammo, "Validation of a Computer Performance Model of the Exponential Queueuing Network Family," Proc. Intl. Symp. on Computer Performance Modeling, Measurement and Evaluation, Cambridge, Mass., March 1976.

[G2] H. Greenberg, Integer Programming, Academic Press, New York, p. 99, 1971.

[K1] T. W. Keller, ASQ Manual, Dept. of Computer Sciences Report TR-27, University of Texas, Austin, Tx., 1973.

[L1] T.-C. Lo, "Computer Aids in Computer Systems Analysis," Ph. D. Thesis, University of Texas, Austin, Tx., 1973.

[M1] R. R. Muntz, "Analytic Modeling of Interactive Systems," Proc. IEEE, Vol. 63, No. 6, pp. 946-953, June 1975.

[R1] M. Reiser and H. Kobayashi, "Queueing Networks with Multiple Closed Chains: Theory and Computational Algorithms," IBM J. Res. Develop., pp. 283-294, May 1975.

[R2] C. A. Rose, "Validation of a Queueing Model with Classes of Customers," Proc. Intl. Symp. on Computer Performance Modeling, Measurement, and Evaluation, Cambridge, Mass., March 1976.

[S1] S. W. Sherman, F. Baskett, and J. C. Browne, "Trace Driven Modeling and Analysis of CPU Scheduling in a Multiprogramming System," Proc. of ACM Workshop on Performance Evaluation, pp. 173-199, Cambridge, Mass., April 1971.

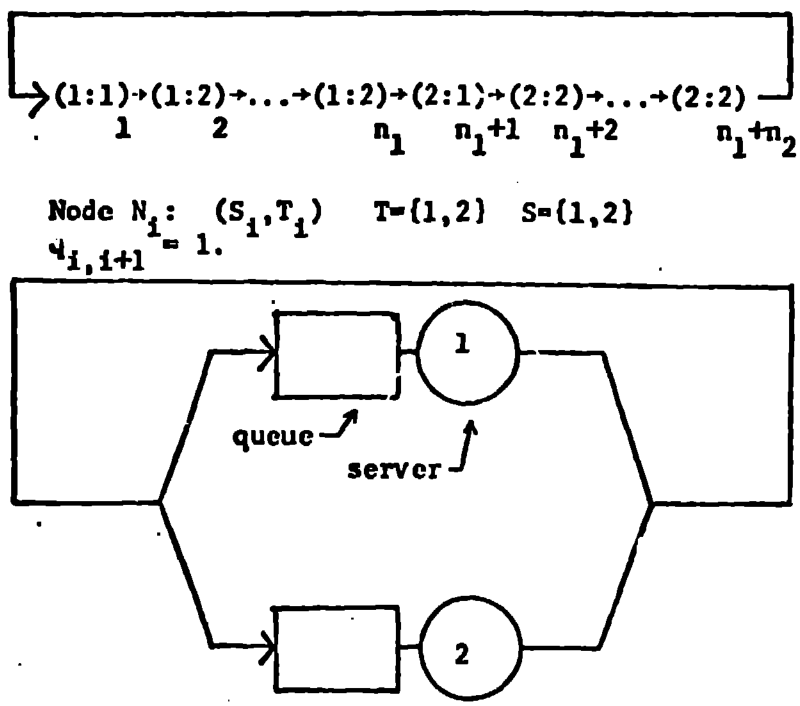
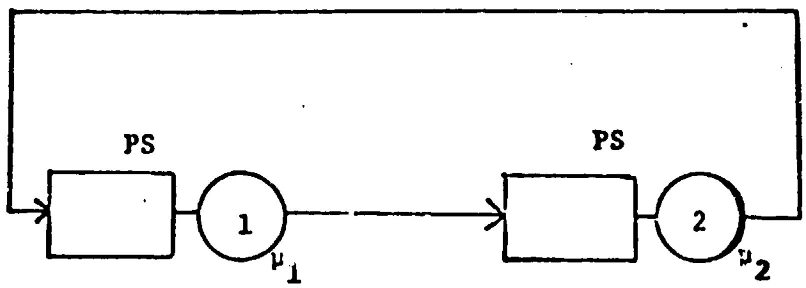


Figure 1: A Customer Task Graph and Its Associated Queueing Network



$p(1,1:2,1)=1/2$, $p(1,1:2,2)=1/2$, $p(1,2:2,1)=1$, $p(1,2:2,2)=0$

$p(2,1:1,1)=1$, $p(2,1:1,2)=0$, $p(2,2:1,1)=1/3$, $p(2,2:1,2)=2/3$.

$e_{11}=1/2$, $e_{12}=1/2$, $e_{21}=1/4$, $e_{22}=3/4$.

$\mu_{11}=1$, $\mu_{12}=2$, $\mu_{21}=2$, $\mu_{22}=3$.

$\beta_{11}=1/2$, $\beta_{12}=1/4$, $\beta_{21}=1/8$, $\beta_{22}=1/4$.

Figure 2: An Example of CTG Equivalence

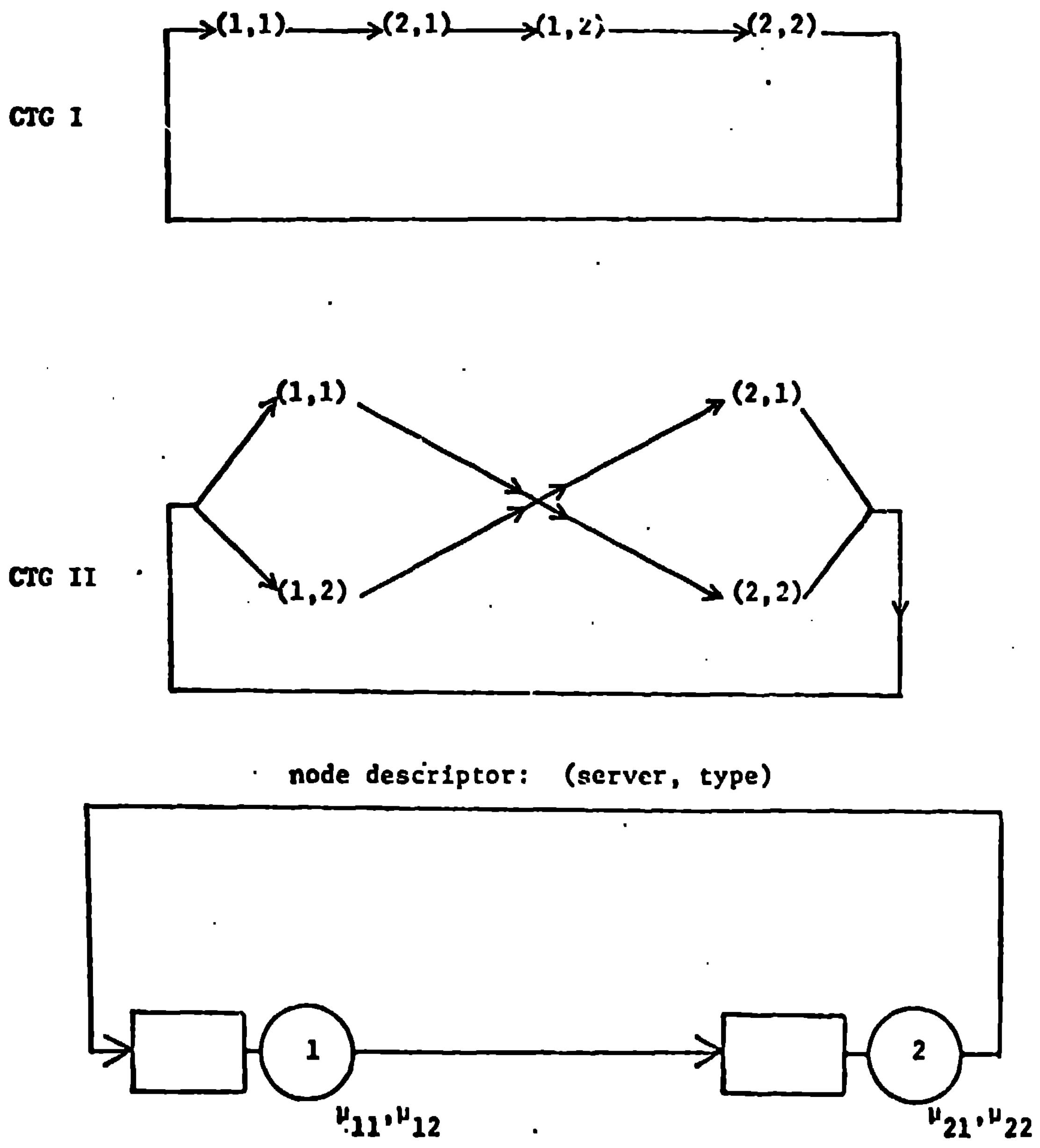
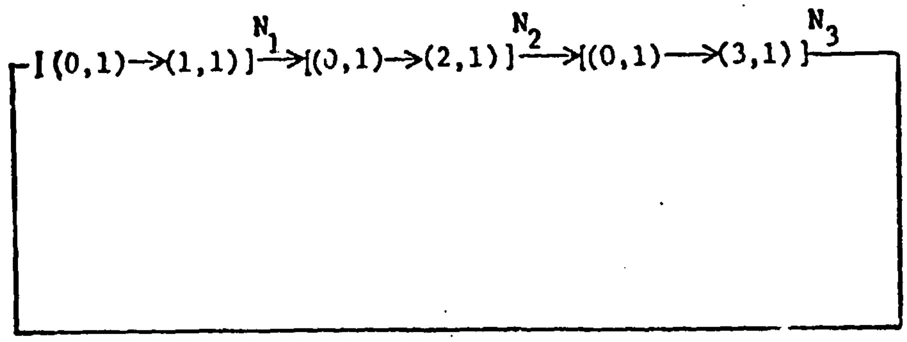
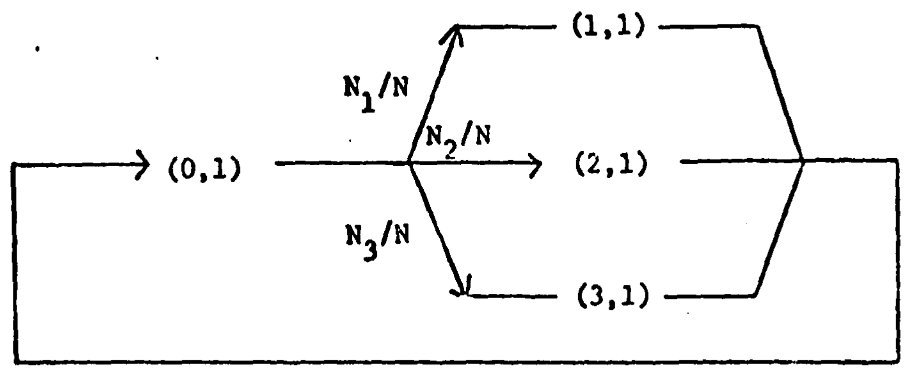


Figure 3: Two Customer Task Graph Models (A)

CTG I



CTG II



$N = N_1 + N_2 + N_3$

$[(i,j)\rightarrow(k,\ell)]^n$ is defined as $(i,j)\rightarrow(k,\ell)_1\rightarrow(i,j)\rightarrow(k,\ell)_2\rightarrow\ldots\rightarrow(k,\ell)_n$

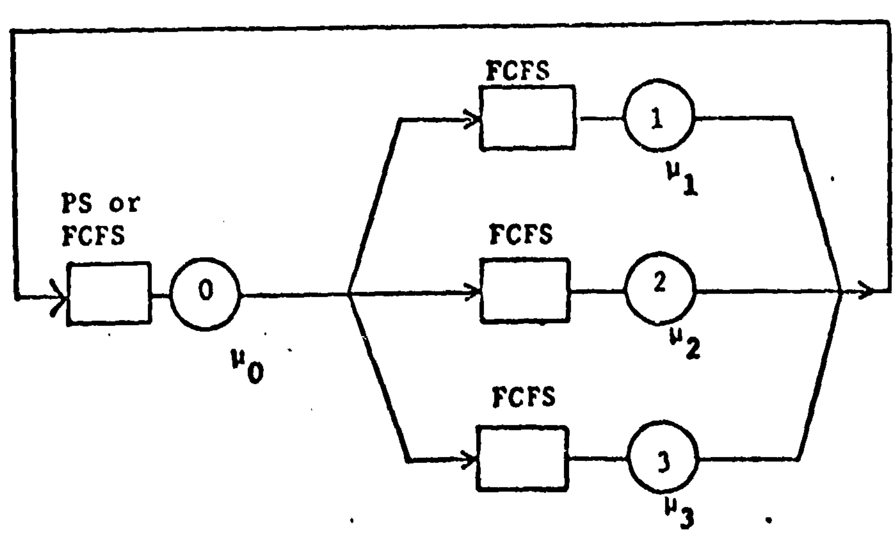


Figure 4: Two Customer Task Graph Models (B)

Table 1: State Probabilities for the Example of Figure 2

State description: (y_1, y_2), $y_1 = (n_{11}, n_{12})$, $y_2 = (n_{21}, n_{22})$
where n_{ij} is the number of customers
of type j in queue i. $M=2$.

n_{11}, n_{12}	n_{21}, n_{22}	$g(y_1)\ g(y_2)$	$g(y_1)\ g(y_2)$
2 0	0 0	β_{11}^2	.25
0 2	0 0	β_{12}^2	.0625
1 1	0 0	$2\beta_{11}\beta_{12}$	.25
1 0	1 0	$\beta_{11}\beta_{21}$	.0625
1 0	0 1	$\beta_{11}\beta_{22}$	.125
0 1	1 0	$\beta_{12}\beta_{21}$	.03125
0 1	0 1	$\beta_{12}\beta_{22}$	.0625
0 0	2 0	β_{21}^2	.015625
0 0	1 1	$2\beta_{21}\beta_{22}$	.0625
0 0	0 2	β_{22}^2	.0625

$G = .984375$

Aggregate state probabilities: $P(n_1, n_2)$

n_1	n_2	$P(n_1, n_2)$
2	0	.5625
0	2	.140625
1	1	.28125

Equivalent network $\beta_1 = \beta_{11} + \beta_{12} = 3/4$, $\beta_2 = \beta_{21} + \beta_{22} = 1/4$.

n_1	n_2	$f(n_1)\ f(n_2)$
2	0	.5625
0	2	.140625
1	1	.28125

Table 2: Comparison of Equivalency
For Model 1, CTG I Vs. CTG II

Service Disciplines: Server 1 - PS, Server 2 - FCFS
P(I): Probability of I Jobs in Queue 1
Number of Customers: 3

CTG	μ_{11}	μ_{12}	μ_{21}	μ_{22}	P(0)	P(1)	P(2)
I	1.0	1.0	1.0	4.0	.216	.275	.510
II					.216	.275	.510
I	1.0	4.0	1.0	4.0	.371	.258	.371
II					.371	.258	.371
I	1.0	16.0	1.0	4.0	.443	.212	.345
II					.443	.212	.345
I	4.0	1.0	1.0	4.0	.371	.258	.371
II					.371	.258	.371
I	4.0	4.0	1.0	4.0	.649	.228	.123
II					.649	.228	.123
I	4.0	16.0	1.0	4.0	.768	.161	.071
II					.768	.161	.071
I	16.0	1.0	1.0	4.0	.443	.212	.345
II					.443	.212	.345
I	16.0	4.0	1.0	4.0	.768	.161	.071
II					.768	.161	.071
I	16.0	16.0	1.0	4.0	.901	.086	.013
II					.901	.086	.013
I	1.0	1.0	1.0	16.0	.201	.223	.576
II					.201	.223	.576
I	1.0	4.0	1.0	16.0	.345	.212	.443
II					.345	.212	.443
I	1.0	16.0	1.0	16.0	.410	.180	.410
II					.410	.180	.410
I	4.0	1.0	1.0	16.0	.345	.212	.443
II					.345	.212	.443
I	4.0	4.0	1.0	16.0	.619	.190	.190
II					.619	.190	.190
I	4.0	16.0	1.0	16.0	.741	.141	.118
II					.741	.141	.118
I	16.0	1.0	1.0	16.0	.410	.180	.410
II					.410	.180	.410
I	16.0	4.0	1.0	16.0	.741	.141	.118
II					.741	.141	.118
I	16.0	16.0	1.0	16.0	.886	.082	.032
II					.886	.082	.032

Table 3: Comparison of Equivalency
For Model 2, CTG I Vs. CTG II

Service Disciplines: Server 1 - PS, Server 2 - FCFS
P(I): Probability of I Jobs in Queue 1
Number of Customers: 2

CTG	μ_{11}	μ_{12}	μ_{21}	μ_{22}	P(0)	P(1)	P(2)	P(3)
I	1.0	1.0	1.0	4.0	.134	.157	.250	.459
II					.135	.158	.247	.460
I	1.0	4.0	1.0	4.0	.260	.238	.243	.260
II					.274	.220	.231	.274
I	1.0	16.0	1.0	4.0	.309	.275	.228	.187
II					.332	.239	.215	.214
I	4.0	1.0	1.0	4.0	.284	.207	.224	.284
II					.274	.220	.231	.274
I	4.0	4.0	1.0	4.0	.623	.212	.109	.057
II					.623	.211	.108	.058
I	4.0	16.0	1.0	4.0	.753	.182	.051	.014
II					.755	.170	.056	.019
I	16.0	1.0	1.0	4.0	.348	.214	.205	.233
II					.332	.239	.215	.214
I	16.0	4.0	1.0	4.0	.756	.161	.060	.023
II					.755	.170	.056	.019
I	16.0	16.0	1.0	4.0	.900	.085	.013	.002
II					.900	.085	.013	.002
I	1.0	1.0	1.0	16.0	.124	.129	.211	.535
II					.126	.131	.208	.536
I	1.0	4.0	1.0	16.0	.228	.201	.228	.343
II					.249	.182	.208	.361
I	1.0	16.0	1.0	16.0	.258	.244	.240	.258
II					.294	.205	.208	.294
I	4.0	1.0	1.0	16.0	.262	.166	.199	.373
II					.249	.182	.208	.361
I	4.0	4.0	1.0	16.0	.585	.168	.128	.119
II					.587	.167	.123	.123
I	4.0	16.0	1.0	16.0	.717	.172	.075	.036
II					.721	.147	.078	.053
I	16.0	1.0	1.0	16.0	.318	.178	.187	.318
II					.294	.205	.208	.294
I	16.0	4.0	1.0	16.0	.724	.133	.081	.062
II					.721	.147	.078	.053
I	16.0	16.0	1.0	16.0	.884	.078	.028	.010
II					.884	.078	.028	.011

Table 4: Comparison of Equivalency
For Model 3, CTG I Vs. CTG II

Service Disciplines: Server 1 - PS, Server 2 - FCFS
P(I): Probability of I Jobs in Queue 1
Number of Customers: 3

CTG	μ_{11}	μ_{12}	μ_{21}	μ_{22}	P(0)	P(1)	P(2)	P(3)
I	1.0	1.0	1.0	4.0	.127	.155	.264	.455
II					.135	.158	.247	.460
I	1.0	4.0	1.0	4.0	.284	.149	.284	.284
II					.302	.198	.198	.302
I	1.0	16.0	1.0	4.0	.362	.130	.259	.249
II					.385	.179	.159	.277
I	4.0	1.0	1.0	4.0	.284	.284	.149	.284
II					.302	.198	.198	.302
I	4.0	4.0	1.0	4.0	.620	.216	.115	.050
II					.623	.211	.108	.058
I	4.0	16.0	1.0	4.0	.755	.136	.087	.022
II					.757	.156	.059	.028
I	16.0	1.0	1.0	4.0	.362	.309	.081	.249
II					.385	.179	.159	.277
I	16.0	4.0	1.0	4.0	.755	.189	.034	.022
II					.757	.156	.059	.028
I	16.0	16.0	1.0	4.0	.900	.086	.013	.001
II					.900	.085	.013	.002
I	1.0	1.0	1.0	16.0	.113	.121	.237	.529
II					.126	.131	.203	.536
I	1.0	4.0	1.0	16.0	.249	.081	.309	.362
II					.277	.159	.179	.385
I	1.0	16.0	1.0	16.0	.317	.048	.317	.317
II					.352	.148	.148	.352
I	4.0	1.0	1.0	16.0	.249	.259	.130	.362
II					.277	.159	.179	.385
I	4.0	4.0	1.0	16.0	.574	.173	.160	.094
II					.587	.167	.123	.123
I	4.0	16.0	1.0	16.0	.719	.085	.154	.043
II					.726	.130	.077	.067
I	16.0	1.0	1.0	16.0	.317	.317	.048	.317
II					.352	.148	.148	.352
I	16.0	4.0	1.0	16.0	.719	.197	.042	.043
II					.726	.130	.077	.067
I	16.0	16.0	1.0	16.0	.883	.081	.031	.004
II					.884	.078	.028	.011

TABLE 5: Comparison of CTG Models with Branching Patterns

Model	M	μ_1^{-1}	μ_2^{-1}, μ_3^{-1}	C_v	ρ	N_1	N_2	N_3	Server O Utilization
1	4	.429	1.29	0	1	1	1	1	.98
						10	10	10	.97
						-	-	-	.97
2	4	.429	1.29	.58	1	1	1	1	.96
						10	10	10	.94
						-	-	-	.95
3	2	.858	2.58	0	2	1	1	1	.58
						10	10	10	.56
						-	-	-	.55
	4	.858	2.58	0	2	1	1	1	.84
						10	10	10	.80
						-	-	-	.79
4	2	.858	2.58	.58	2	1	1	1	.56
						10	10	10	.53
						-	-	-	.55
	4	.858	2.58	.58	2	1	1	1	.79
						10	10	10	.77
						-	-	-	.76
5	4	.556	1.67	.58	1	3	1	1	.97
						30	10	10	.95
						-	-	-	.96
6	2	1.12	3.36	.58	2	3	1	1	.58
						30	10	10	.57
						-	-	-	.56
	4	1.12	3.36	.58	2	3	1	1	.86
						30	10	10	.81
						-	-	-	.83
7	2	1.12	3.36	.58	2	3	1	1	.55
						30	10	10	.55
						-	-	-	.55
	4	1.12	3.36	.58	2	3	1	1	.80
						30	10	10	.79
						-	-	-	.79
8	4	.429	1.29	.58	1	1	1	1	.96
						10	10	10	.96
						-	-	-	.96

Model	M	μ_1^{-1}	μ_2^{-1}, μ_3^{-1}	C_v	ρ	N_1	N_2	N_3	Server 0 Utilization
9	2	.858	2.58	.58	2	1	1	1	.56
						10	10	10	.55
						-	-	-	.54
10	2	1.12	3.36	0	2	3	1	1	.56
						30	10	10	.56
						-	-	-	.56
	4	1.12	3.36	0	2	3	1	1	.82
						30	10	10	.80
						-	-	-	.79
11	2	1.12	3.36	.58	2	3	1	1	.59
						30	10	10	.57
						-	-	-	.56

For models 8-11 all queues are disciplined FCFS.